ROPES & GRAY

ROPES & GRAY LLP
1211 AVENUE OF THE AMERICAS
NEW YORK, NY 10036-8704
WWW.ROPESGRAY.COM

Jesse J. Jenner
T +1 212 596 9019
F +1 646 728 2581
jesse.jenner@ropesgray.com

March 15, 2012

BY FEDEX

Honorable James L. Robart
United States District Judge
United States District Court
700 Stewart St., Suite 14128
Seattle, WA 98101-9906

Microsoft Corp. v. Motorola, Inc., et al.,
No. C10-1823-JLR

Dear Judge Robart:

Pursuant to the Court's instruction during the March 9, 2012, *Markman* hearing, I enclose a copy of the narrative that Dr. Timothy Drabik used during his tutorial. Because we now have the transcript of the hearing, we have conformed the narrative notes to the transcript, so that the enclosed narrative reflects the words Dr. Drabik actually used during the hearing.

Also, in response to the Court's "homework assignment" (Hearing transcript, 9:17-22), I enclose some downloads from the internet regarding "Captain Midnight and the Secret Decoder Ring", a classic in its own time.

Respectfully,

Jesse J. Jenner

Jesse J. Jenner

JJJ:epb
Enclosures

cc: Richard A. Cederoth, Esq. (with enclosures)

§1

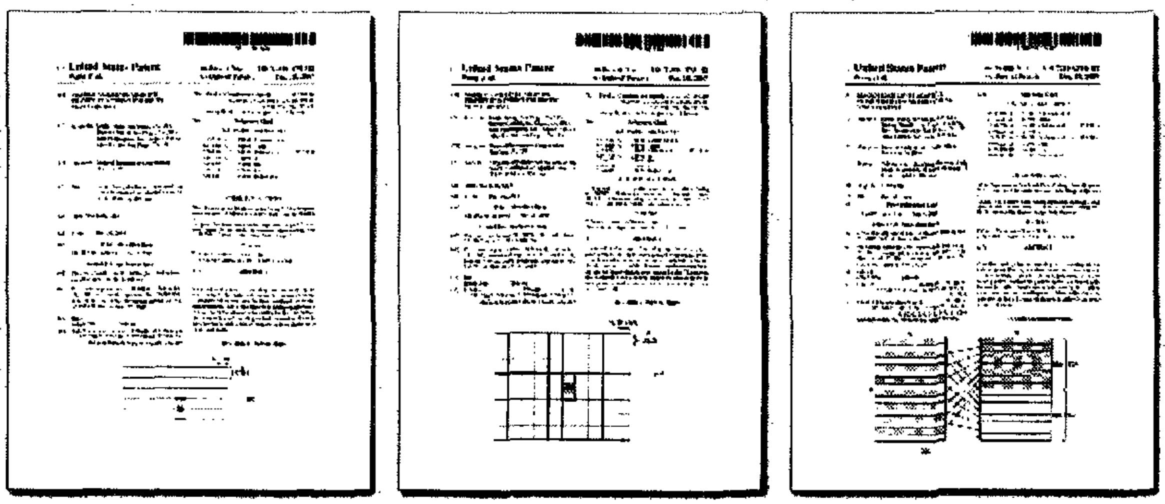
Tutorial of Motorola Regarding U.S. Patents 7,310,374, 7,310,375, and 7,310,376

Microsoft Corp. v. *Motorola, Inc., et al.*

Case No. C10-1823-JLR
(W.D. Wash.)

§2 This relates to the '374 patent, '375 patent and '376 patent. These patents share a common specification, and they all relate to video coding, in particular, the techniques used in the video coding standard called H.264.

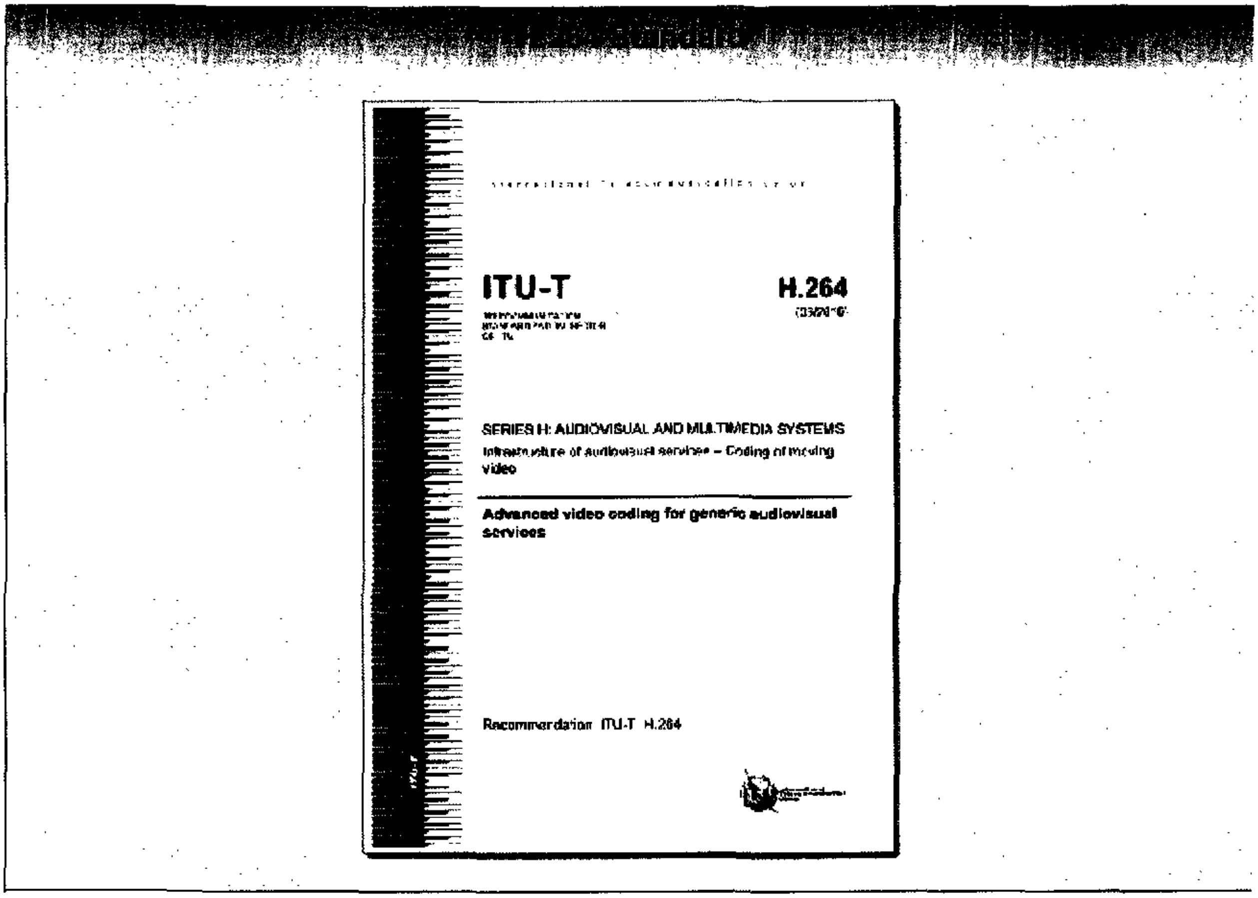
U.S. Patent 7,310,374 **U.S. Patent 7,310,375** **U.S. Patent 7,310,376**



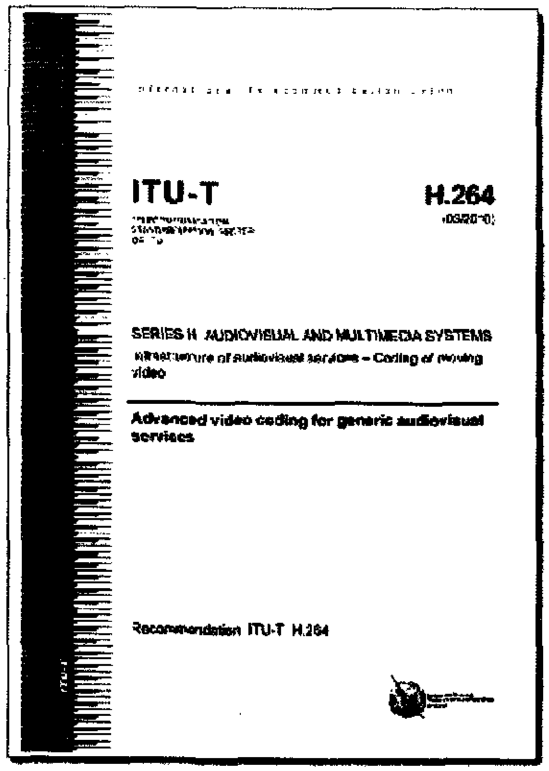
*** The specifications for each of these patents is identical.**
Unless otherwise indicated, citations are to the '374 Patent.

29682505_10

§3,1 Here is the cover picture of the H.264 standard. It is also sometimes referred to as MPEG-4 Part 10, or AVC, which stands for advanced video coding. Motorola inventors on these patents participated in the development of this standard.



§3,2 The draft of this standard is incorporated by reference in the patents in suit. And the H.264 standard was published in 2003.



ITU-T

H.264

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Coding of moving video

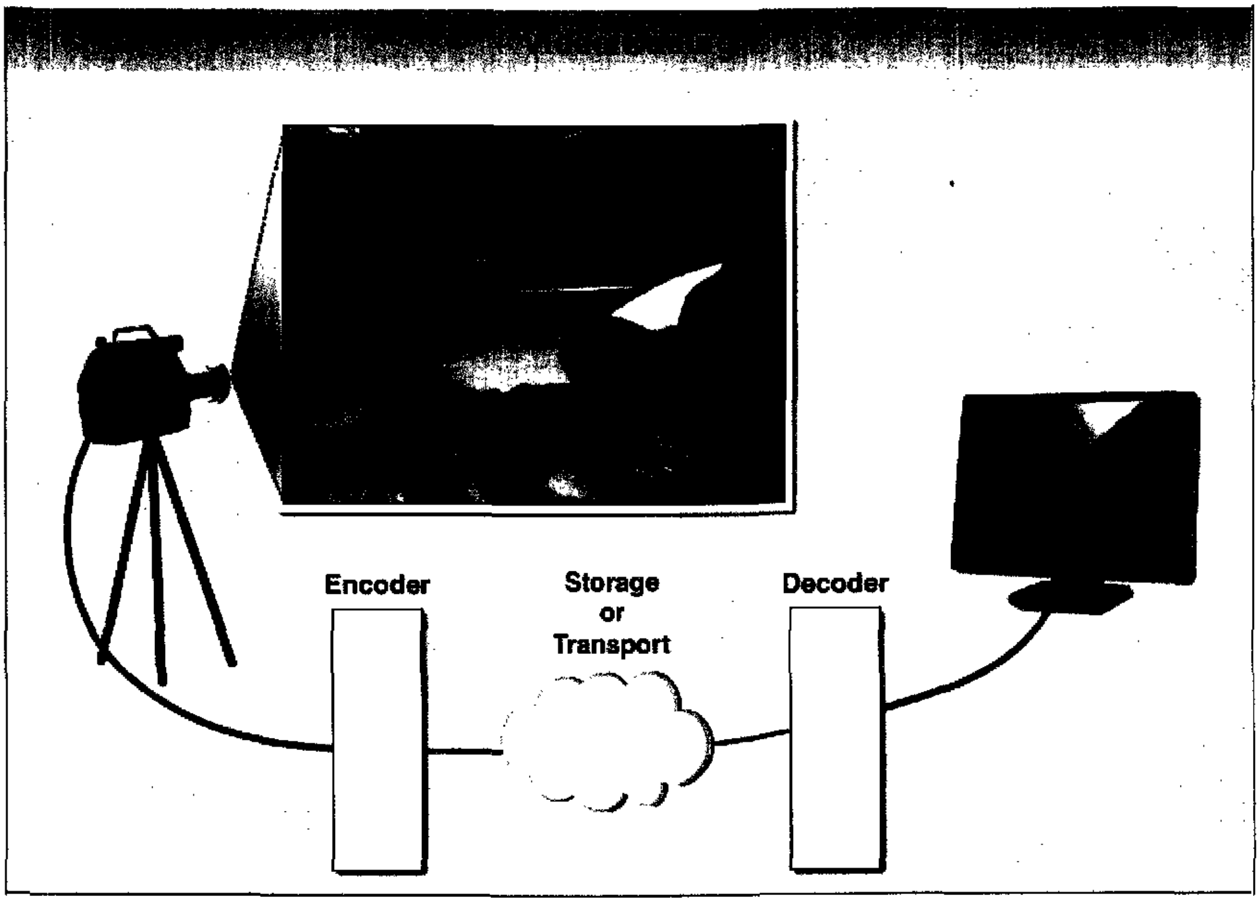
Advanced video coding for generic audiovisual services

Recommendation ITU-T H.264

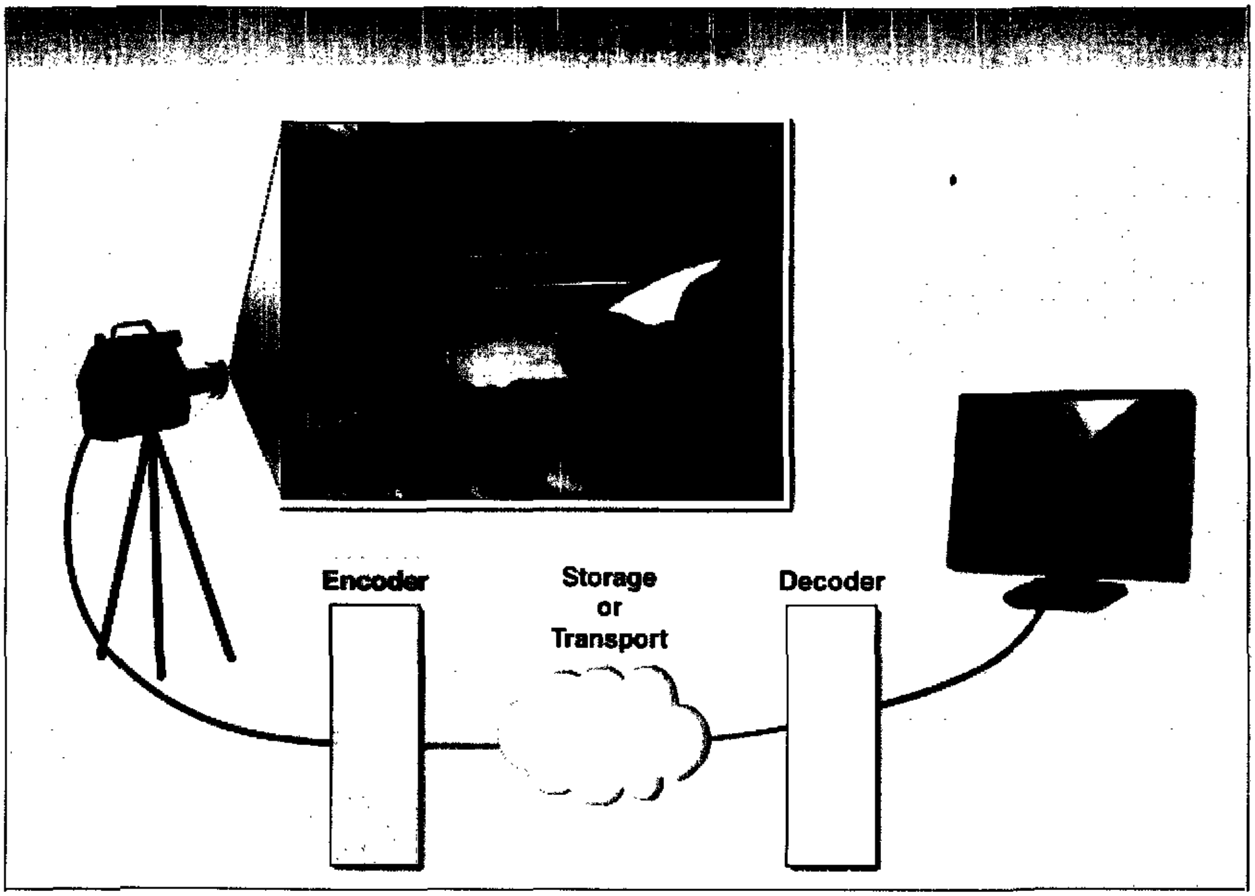
The present invention relates to encoding and decoding of digital video content. More specifically, the present invention relates to frame mode and field mode encoding of digital video content at a macroblock level as used in the MPEG-4 Part 10 AVC/H.264 standard video coding standard.

'374 Patent, 1:16-21

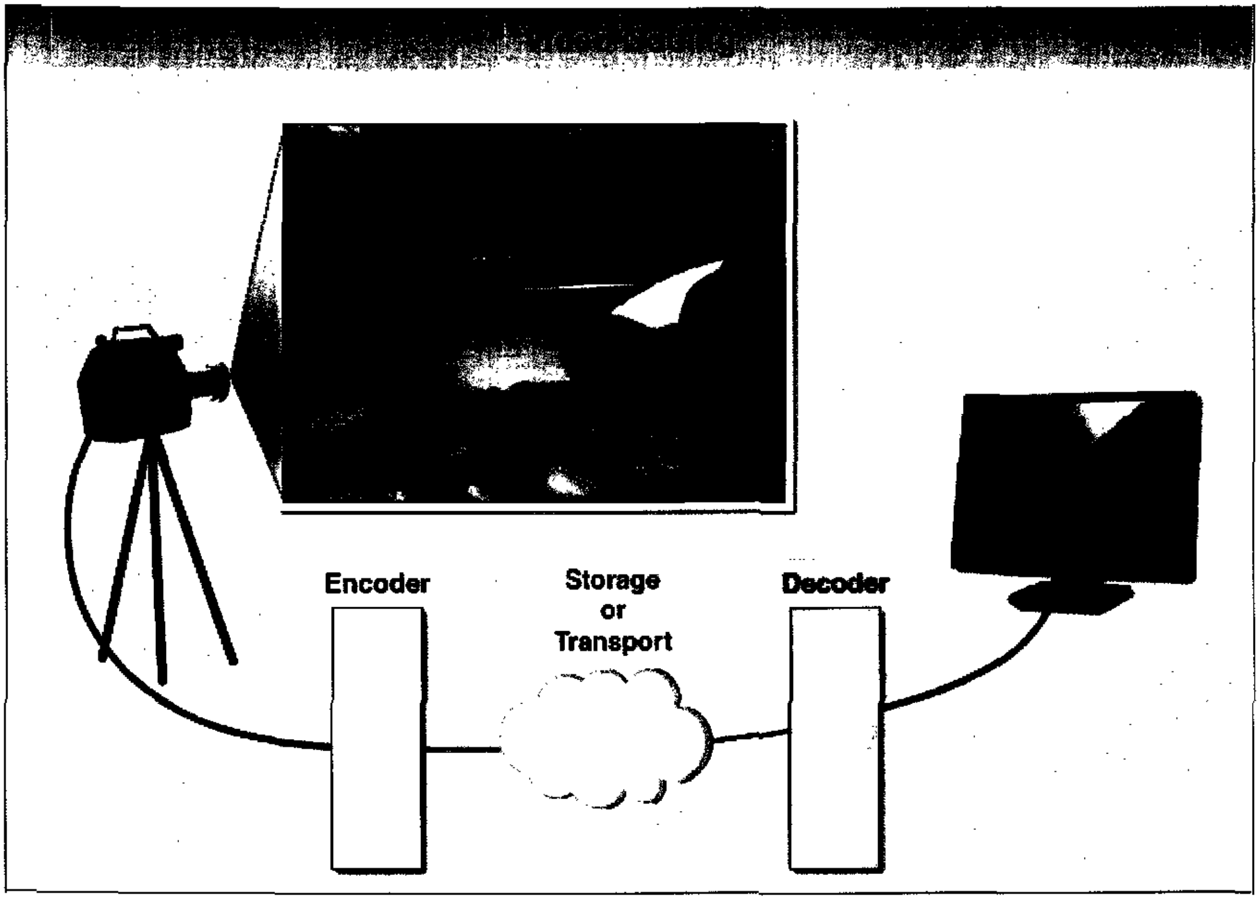
§4,1 The patents and the H.264 standard concerned video coding. Video coding is the process of coding and decoding digital video.



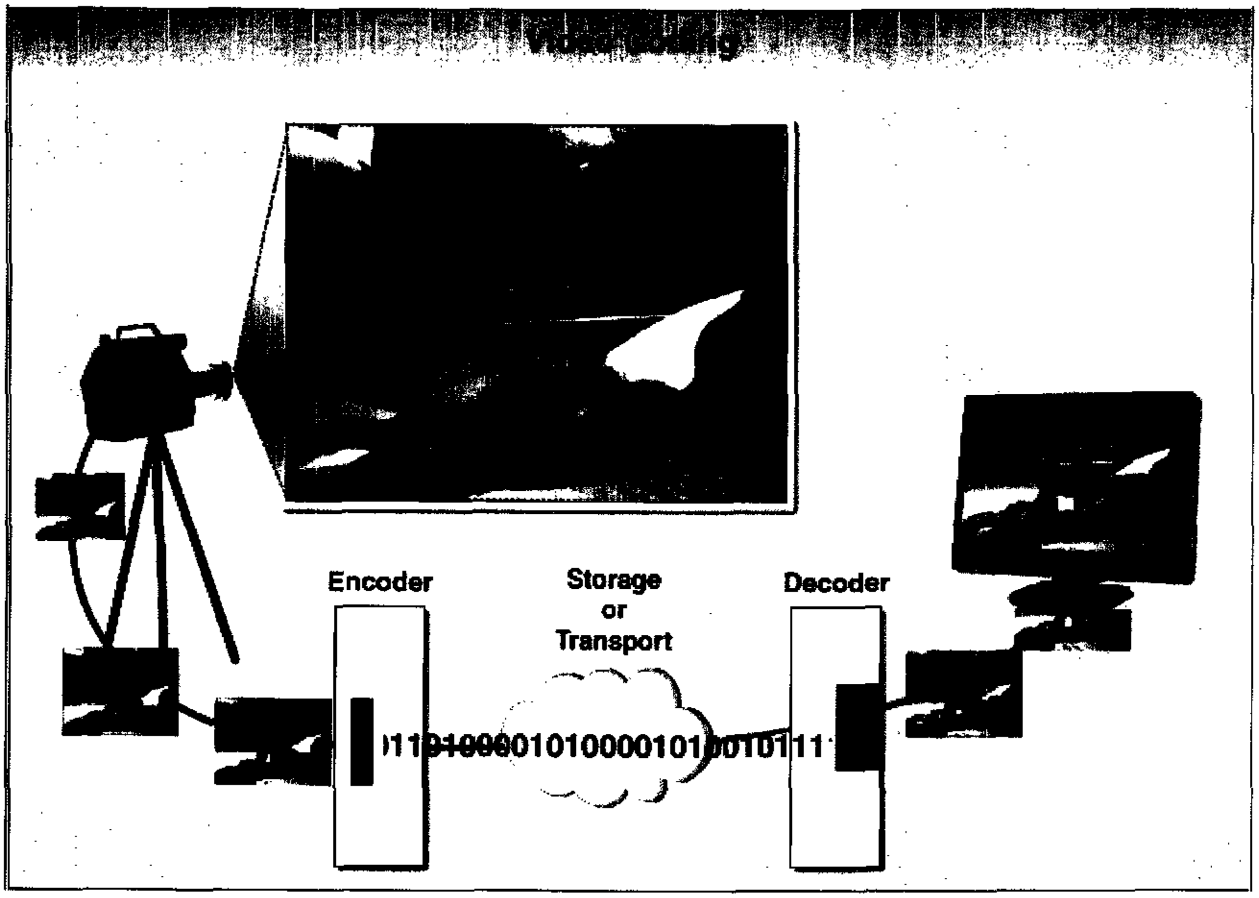
§4,2 Encoding is compressing a stream of pictures,



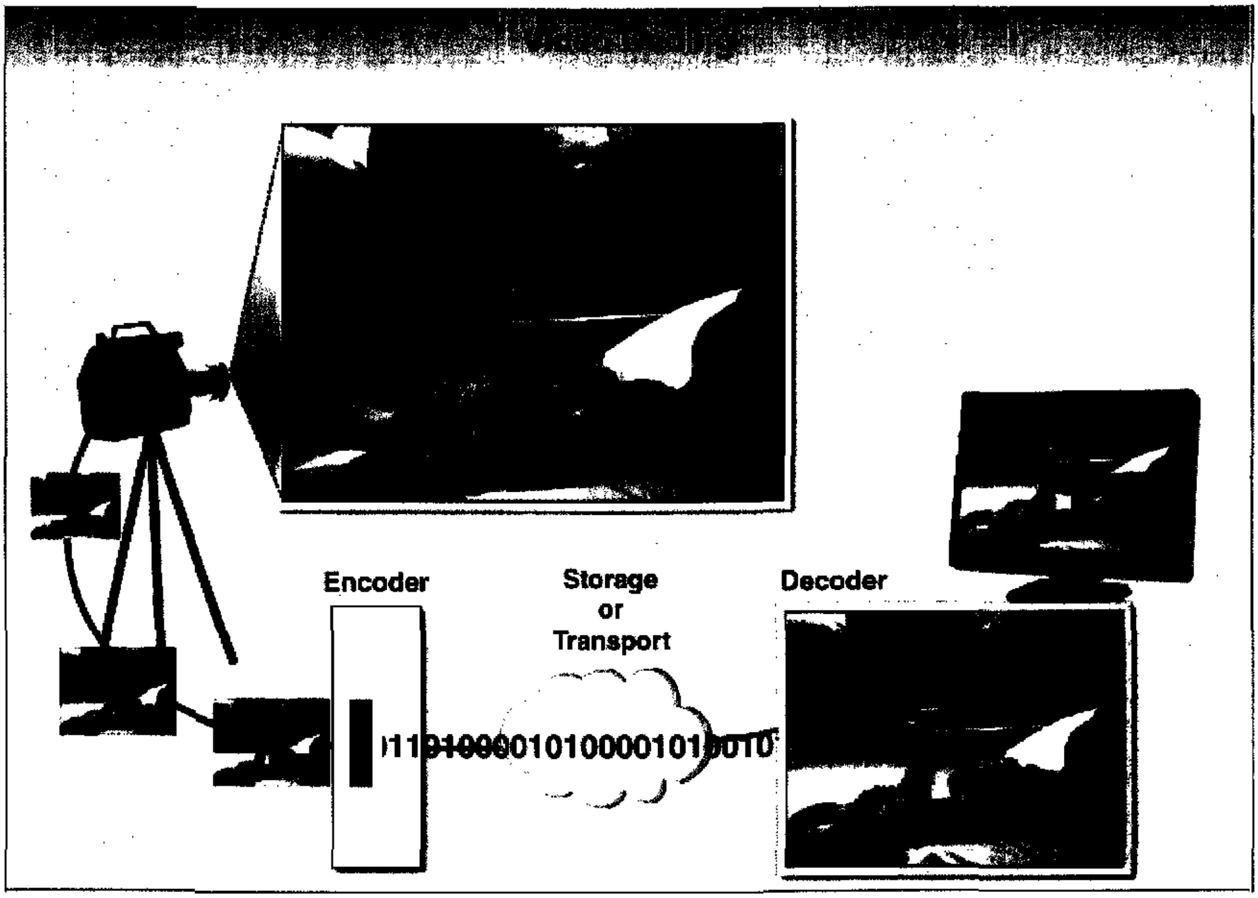
§4,3 and decoding is decompressing a stream of pictures.



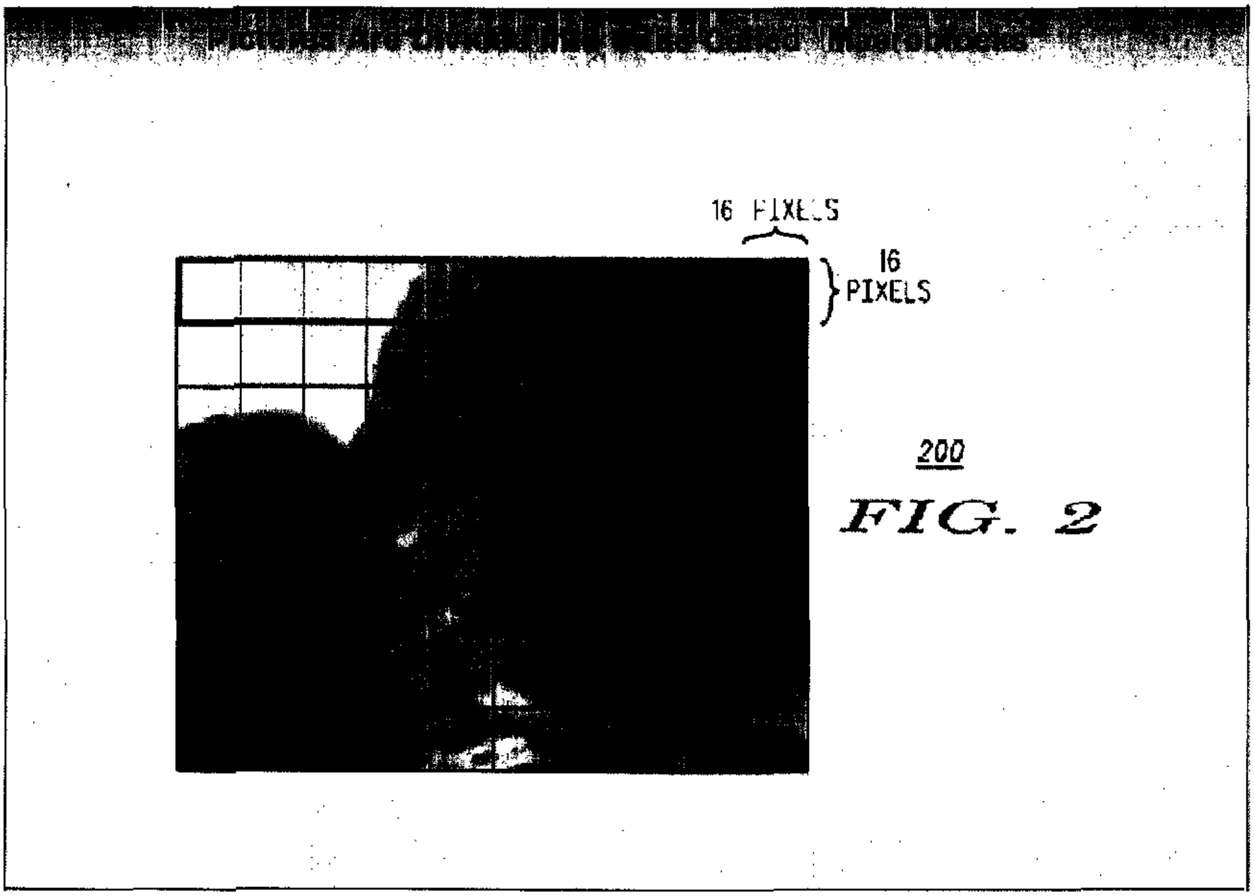
§4,4-195 Here, I will show an example of a video coding process. The video camera on the left captures a stream of pictures and sends it to an encoder. The encoder compresses the stream of video data by eliminating redundancies in the video. The output of the encoder is a bitstream of compressed video data that can be transported and stored, as represented here by the cloud. This compressed video stream takes up much less space in storage than the uncompressed content, or requires much less bandwidth for transmission. After the compressed video data has been transmitted, it must be decoded or decompressed.



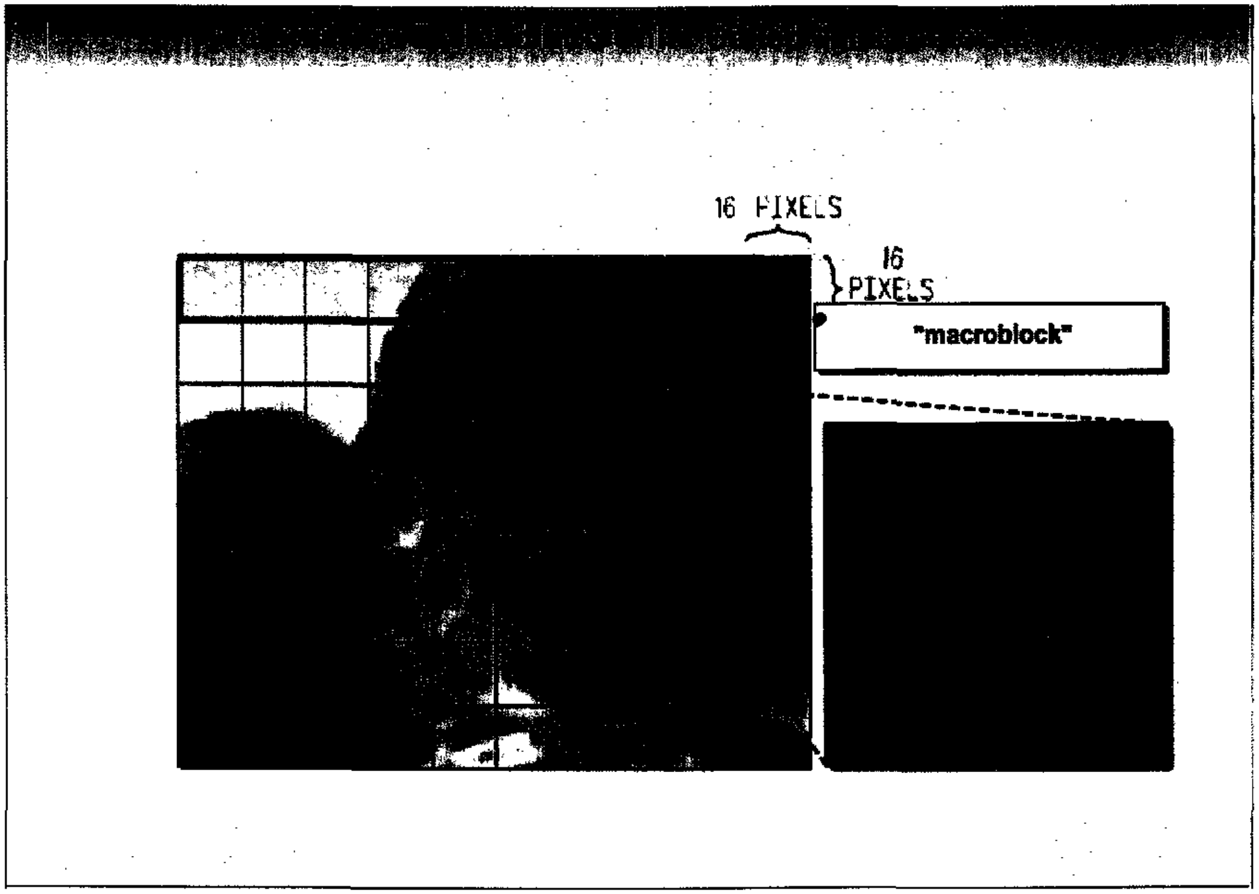
§4,196-294 And here, I am showing the decoder having constructed a decoded picture highlighted in yellow. After decoding, the decoded pictures can be displayed, for example, on a monitor, as I have shown there on the right.



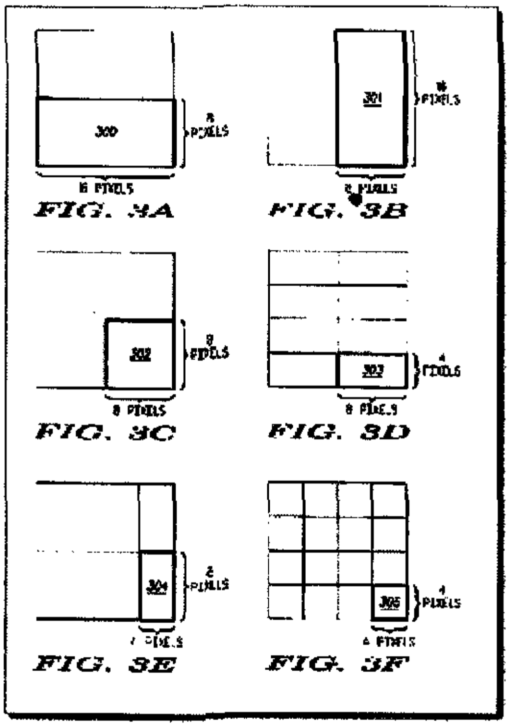
§4,295-634 Now, video coding operates on a stream of pictures. Each picture is divided into basic units called macroblocks. For example, we will see a red box around the baby leopard's eye. That is a macroblock.



§4,635-684 Now, Figure 2 shows the macroblock having 16 by 16 pixels. A pixel is frequently defined as the smallest independent unit of picture information.

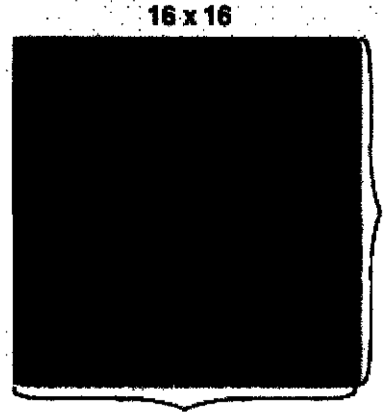


§4,685-727 For a very important step, referred to as prediction, each macroblock can be further divided into smaller-size blocks, as depicted on the left.



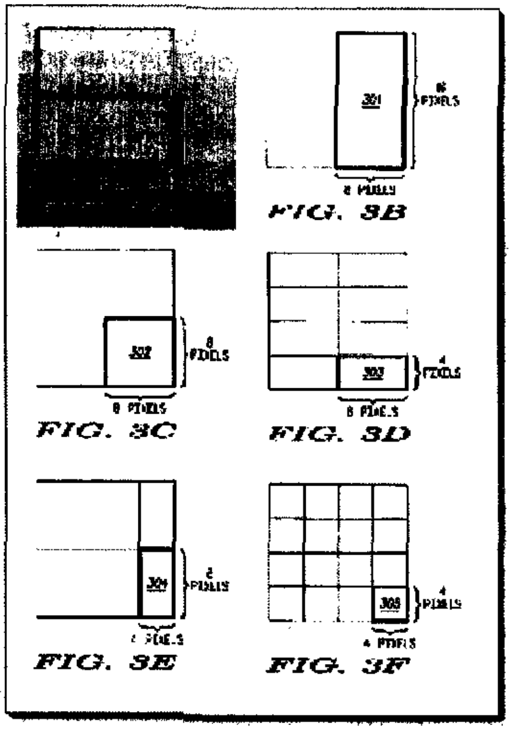
FIGS. 3a-f show that a macroblock can be further divided into small[illegible] For example, as shown in FIGS. 3a-f, a macroblock can be further divided into block sizes of 16 by 8 pixels (FIG. 3a; 300), 8 by 16 pixels (FIG. 3b; 301), 8 by 8 pixels (FIG. 3c; 302), 8 by 4 pixels (FIG. 3d; 303), 4 by 8 pixels (FIG. 3e; 304), or 4 by 4 pixels (FIG. 3f; 305).

'374 Patent, 5:59-64



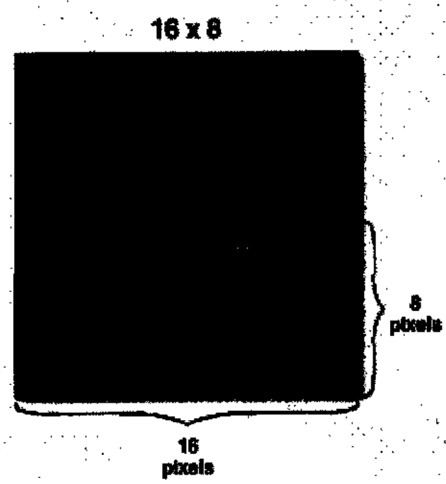
29682505_10

§4,728-738 Here, we see a 16 by 8 block size in the figure, as well as on the macroblock.

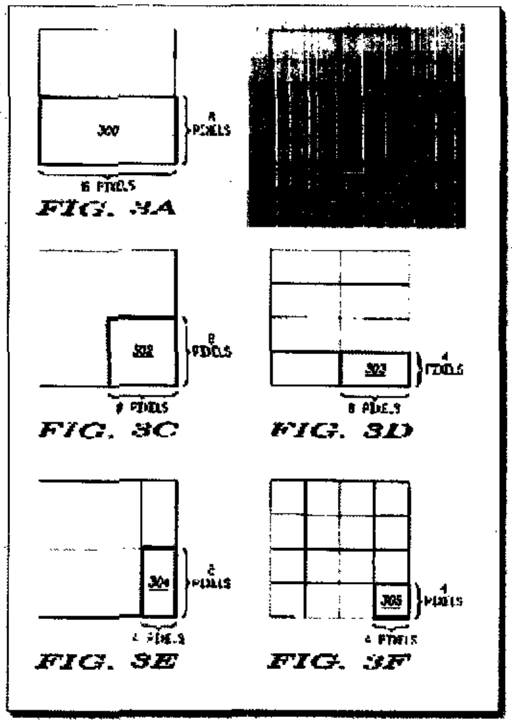


FIGS. 3a-f show that a macroblock can be further divided into smaller sized blocks. For example, as shown in FIGS. 3a-f, a macroblock can be further divided into block sizes of [redacted] (FIG. 3a; 300), 8 by 16 pixels (FIG. 3b; 301), 8 by 8 pixels (FIG. 3c; 302), 8 by 4 pixels (FIG. 3d; 303), 4 by 8 pixels (FIG. 3e; 304), or 4 by 4 pixels (FIG. 3f; 305).

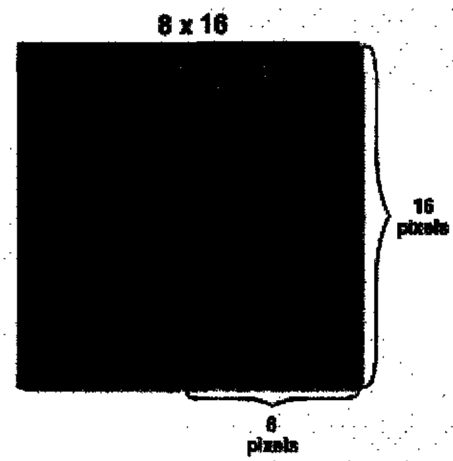
'374 Patent, 5:59-64



§4,739 And now we see an 8 by 16 block size.

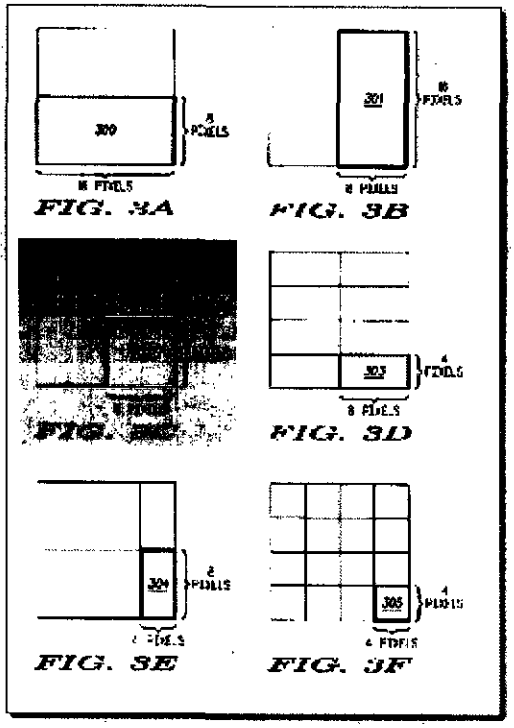


> FIGS. 3a-f show that a macroblock can be further divided into smaller sized blocks. For example, as shown in FIGS. 3a-f, a macroblock can be further divided into block sizes of 16 by 8 pixels (FIG. 3a; 300), [redacted] (FIG. 3b; 301), 8 by 8 pixels (FIG. 3c; 302), 8 by 4 pixels (FIG. 3d; 303), 4 by 8 pixels (FIG. 3e; 304), or 4 by 4 pixels (FIG. 3f; 305).
>
> '374 Patent, 5:59-64



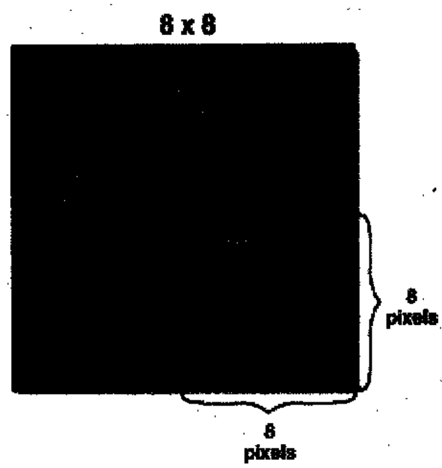
29682505_10

§4,740 Then an 8 by 8 block size,



FIGS. 3a-f show that a macroblock can be further divided into smaller sized blocks. For example, as shown in FIGS. 3a-f, a macroblock can be further divided into block sizes of 16 by 8 pixels (FIG. 3a; 300), 8 by 16 pixels (FIG. 3b; 301),[redacted](FIG. 3c; 302), 8 by 4 pixels (FIG. 3d; 303), 4 by 8 pixels (FIG. 3e; 304), or 4 by 4 pixels (FIG. 3f; 305).

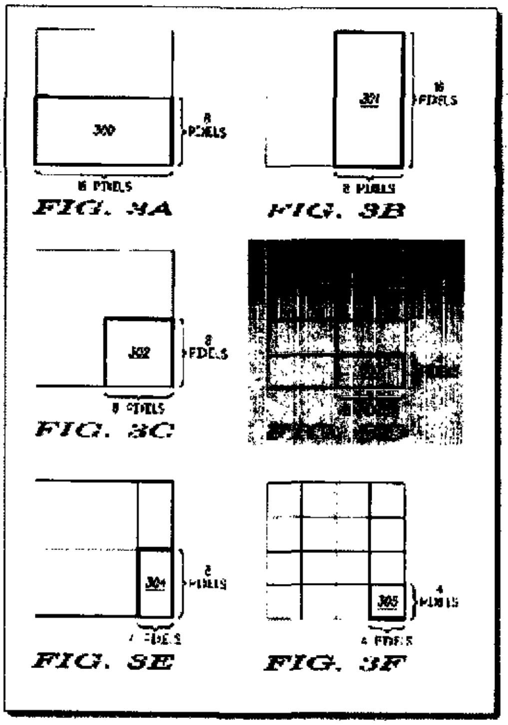
'374 Patent, 5:59-64



29682505_10

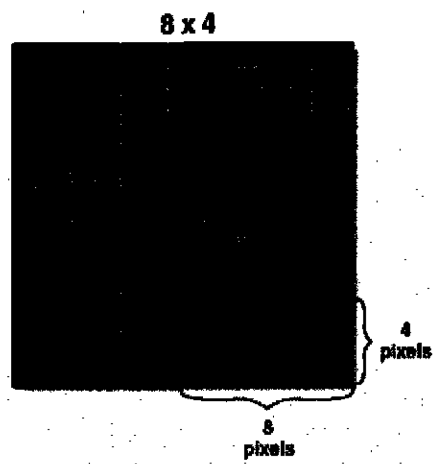
§4,741 8 by 4 block size,

A Macroblock Can Be Divided Into Smaller Sized Blocks



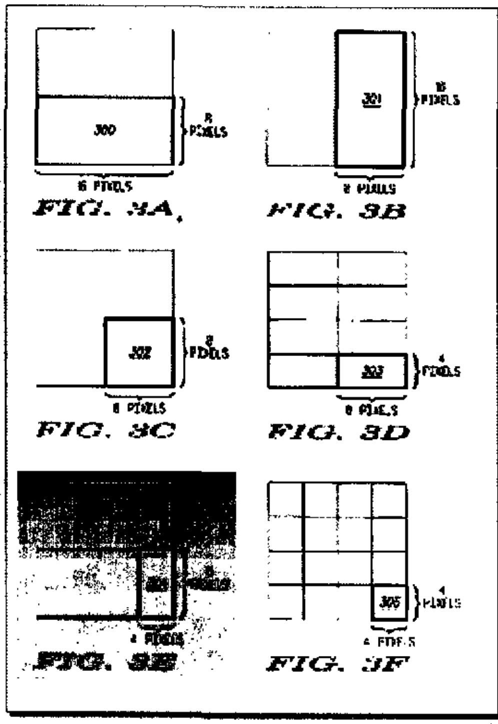
FIGS. 3a-f show that a macroblock can be further divided into smaller sized blocks. For example, as shown in FIGS. 3a-f, a macroblock can be further divided into block sizes of 16 by 8 pixels (FIG. 3a; 300), 8 by 16 pixels (FIG. 3b; 301), 8 by 8 pixels (FIG. 3c; 302), [redacted] (FIG. 3d; 303), 4 by 8 pixels (FIG. 3e; 304), or 4 by 4 pixels (FIG. 3f; 305).

'374 Patent, 5:59-64



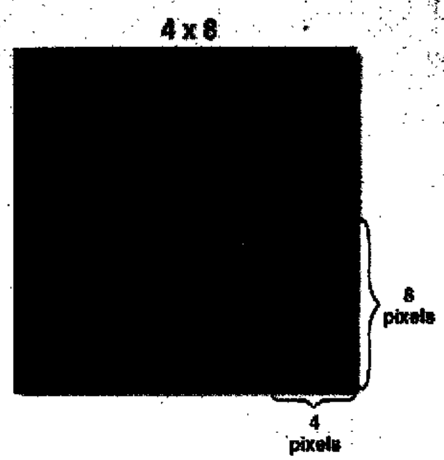
29682505_10

§4,742 4 by 8 block size,



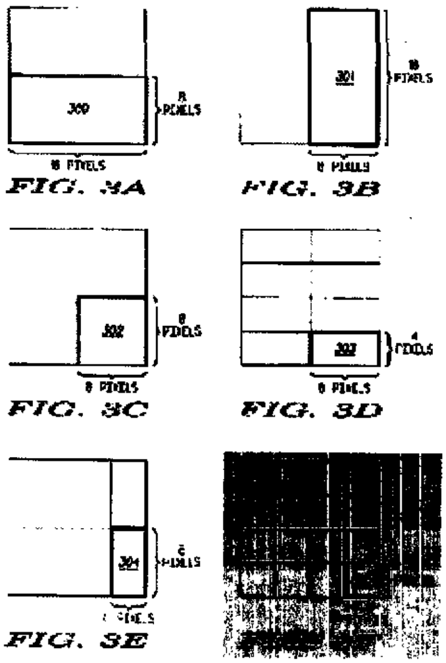
FIGS. 3a-f show that a macroblock can be further divided into smaller sized blocks. For example, as shown in FIGS. 3a-f, a macroblock can be further divided into block sizes of 16 by 8 pixels (FIG. 3a; 300), 8 by 16 pixels (FIG. 3b; 301), 8 by 8 pixels (FIG. 3c; 302), 8 by 4 pixels (FIG. 3d; 303),[redacted](FIG. 3e; 304), or 4 by 4 pixels (FIG. 3f; 305).

'374 Patent, 5:59-64



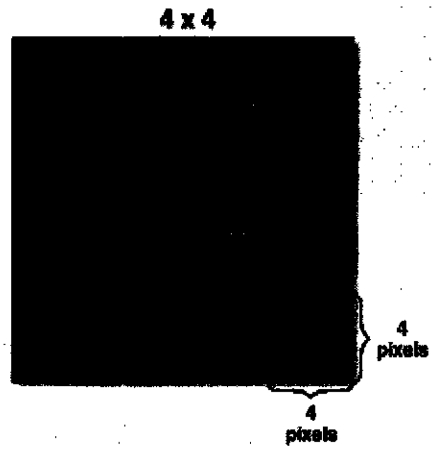
29682505_10

§4,743 and a 4 by 4 block size. These blocks allow us to predict different size portions of the macroblock separately.



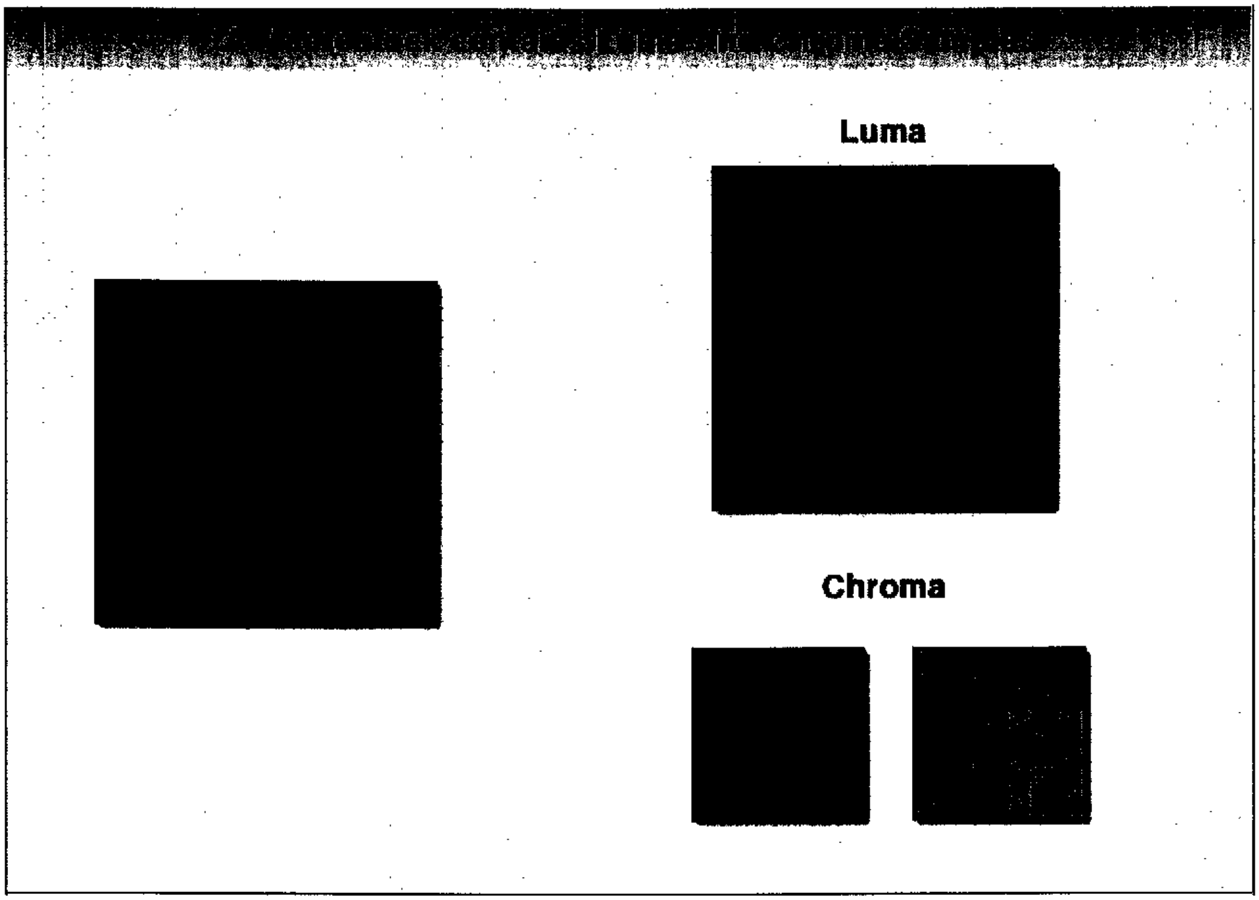
FIGS. 3a-f show that a macroblock can be further divided into smaller sized blocks. For example, as shown in FIGS. 3a-f, a macroblock can be further divided into block sizes of 16 by 8 pixels (FIG. 3a; 300), 8 by 16 pixels (FIG. 3b; 301), 8 by 8 pixels (FIG. 3c; 302), 8 by 4 pixels (FIG. 3d; 303), 4 by 8 pixels (FIG. 3e; 304), or [redacted] (FIG. 3f; 305).

'374 Patent, 5:59-64

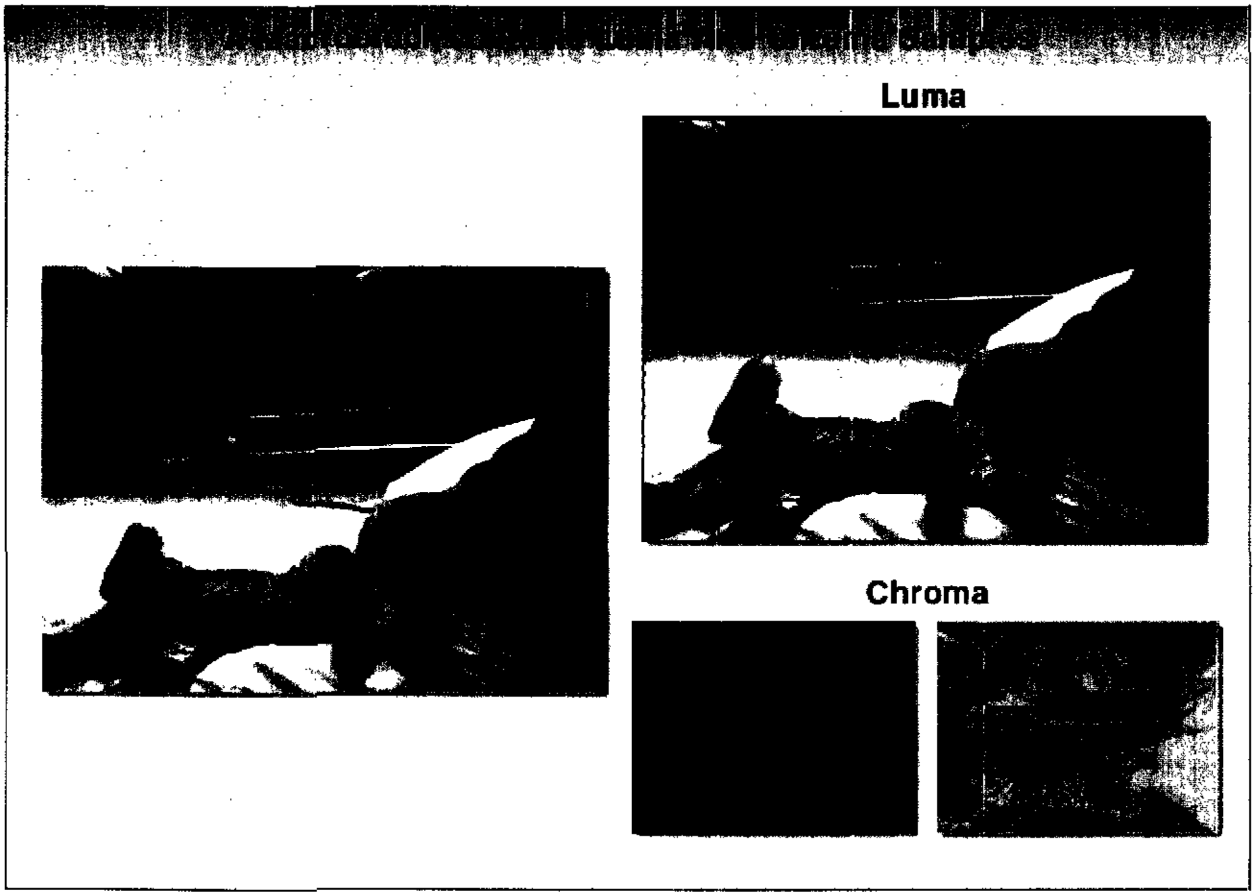


29682505_10

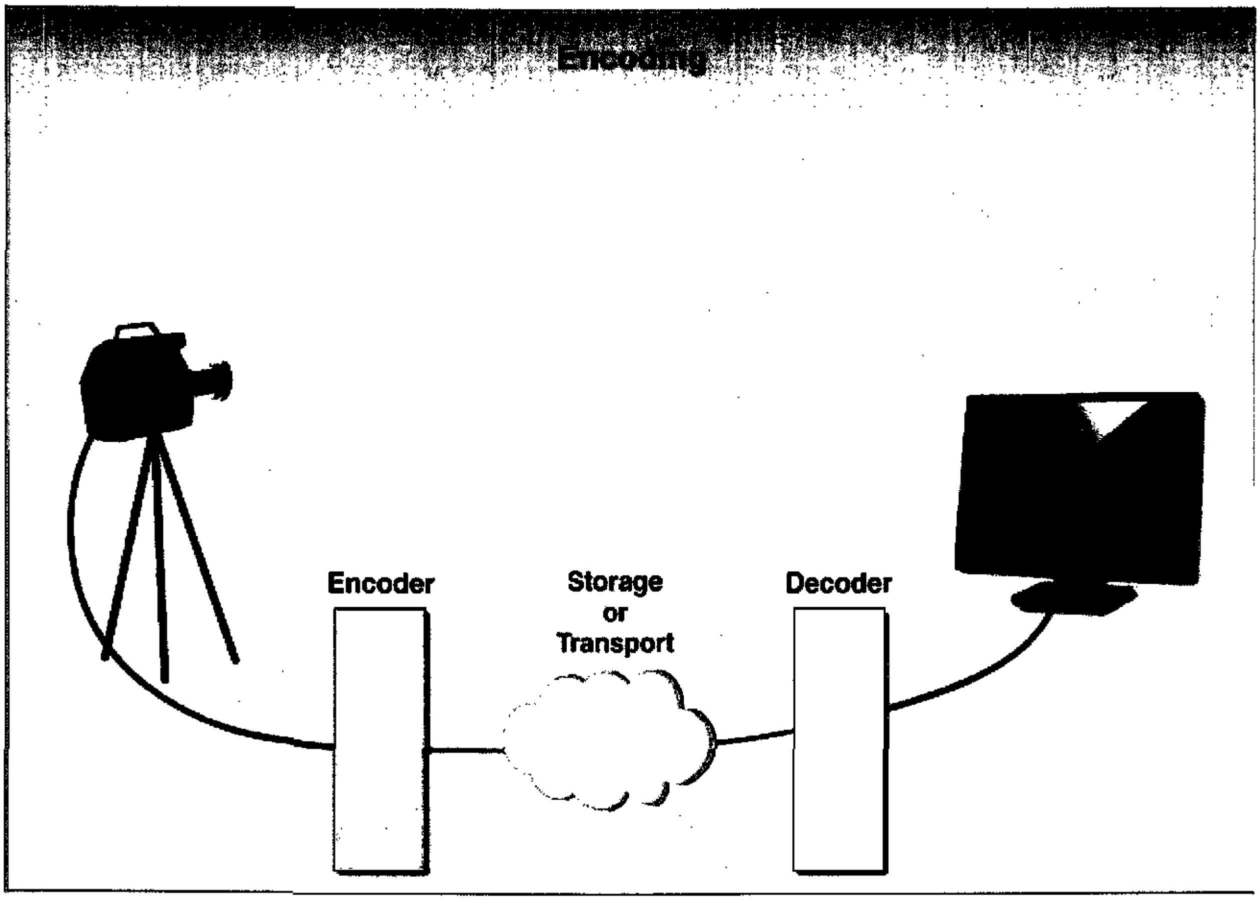
§4,744-923 Now, macroblocks also contain luma and chroma components.



§4,924-1055 The luma components represent the brightness of the picture, as in the black and white picture. The chroma components, and there are two of them, provide color information. Merging all three of these components gives you a color picture. Because the human eye is less sensitive to color changes than to brightness changes, it is common for a macroblock to have fewer chroma samples than luma samples.

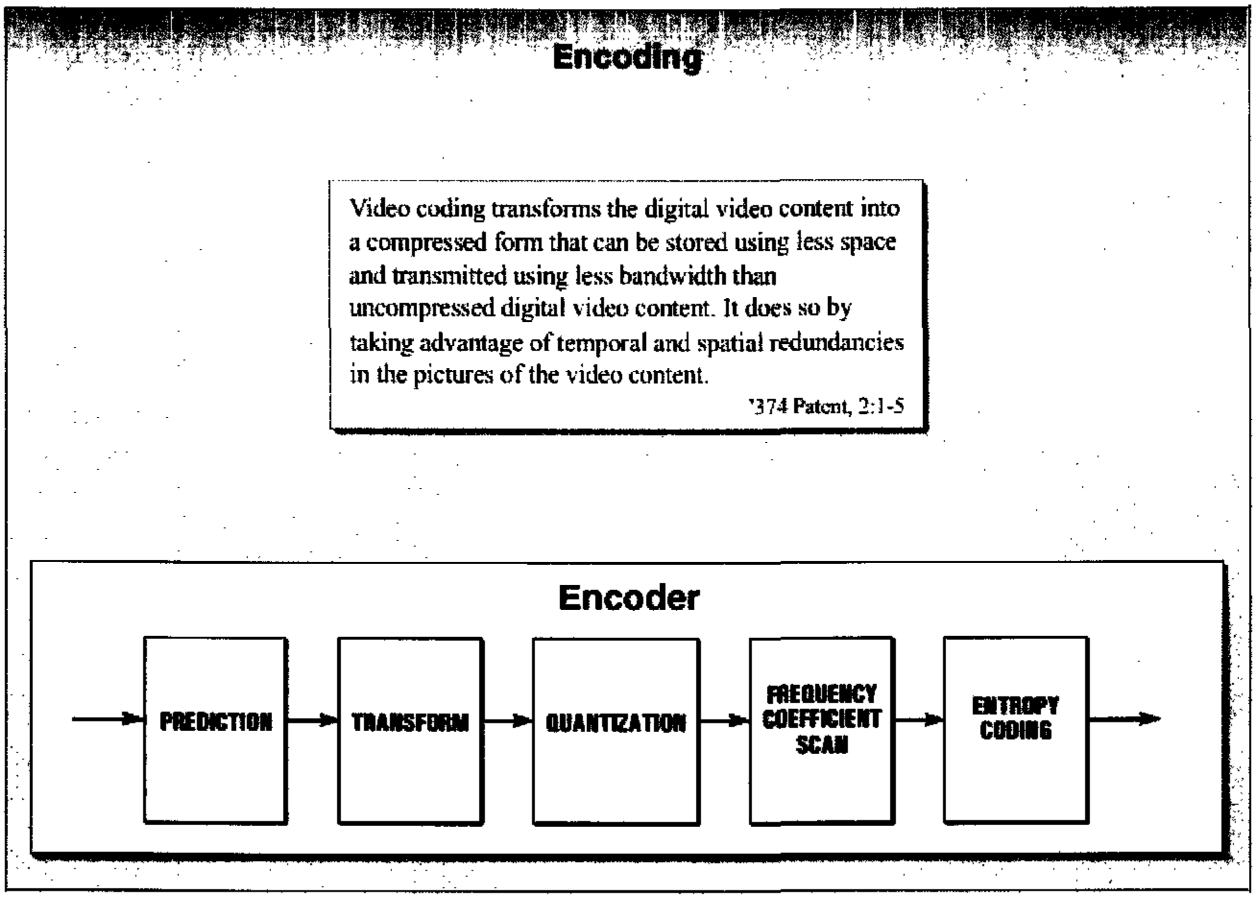


§5,1, I would like to discuss encoding in detail now.

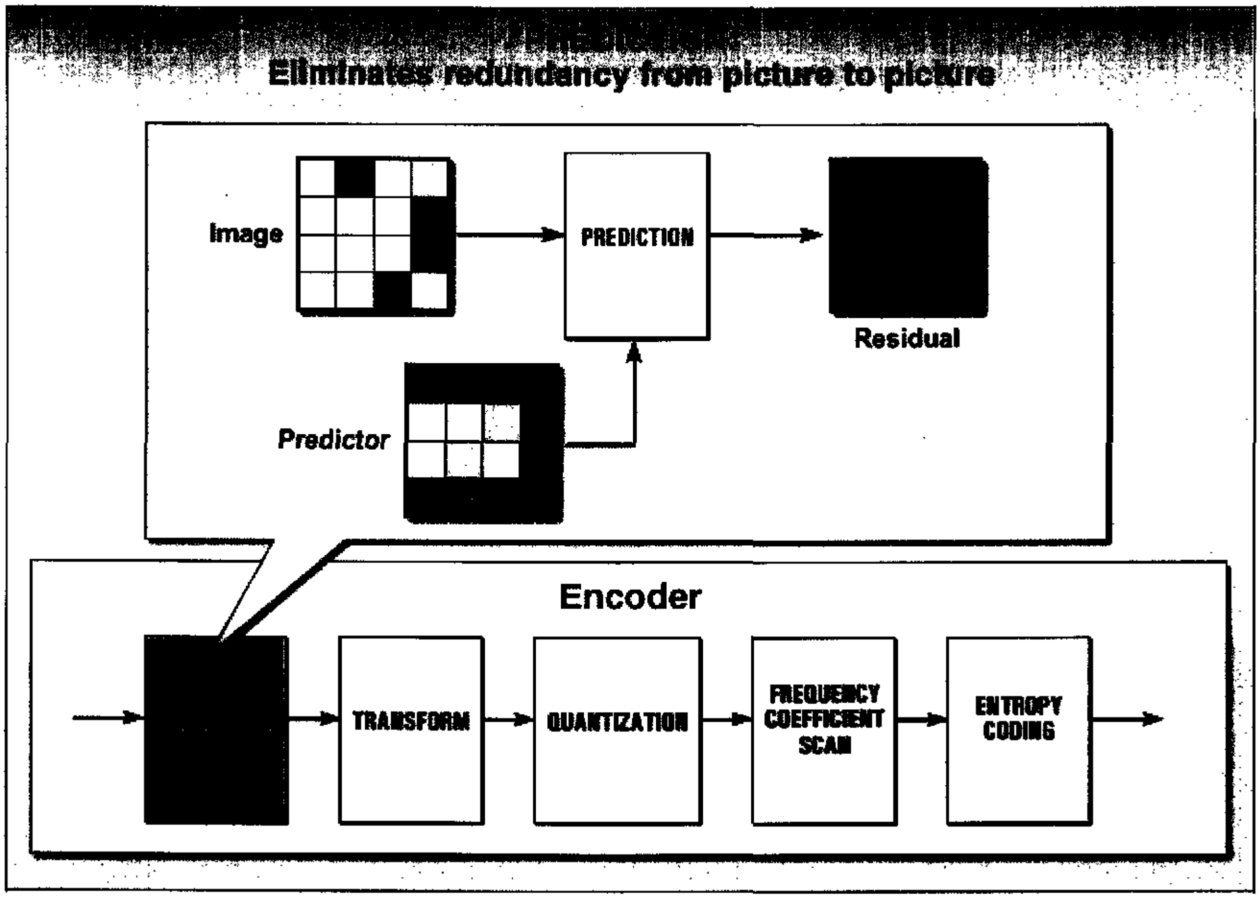


§5,2-75 Here we are back with our encoder. I would like to blow up that box, and show an example of the general process for encoding digital video into the coded bitstream. The coded bitstream is compressed for easier transmission or storage, as I have said. And the encoding operations proceed from left to right, as indicated by the arrows. And the details of each step are well known.

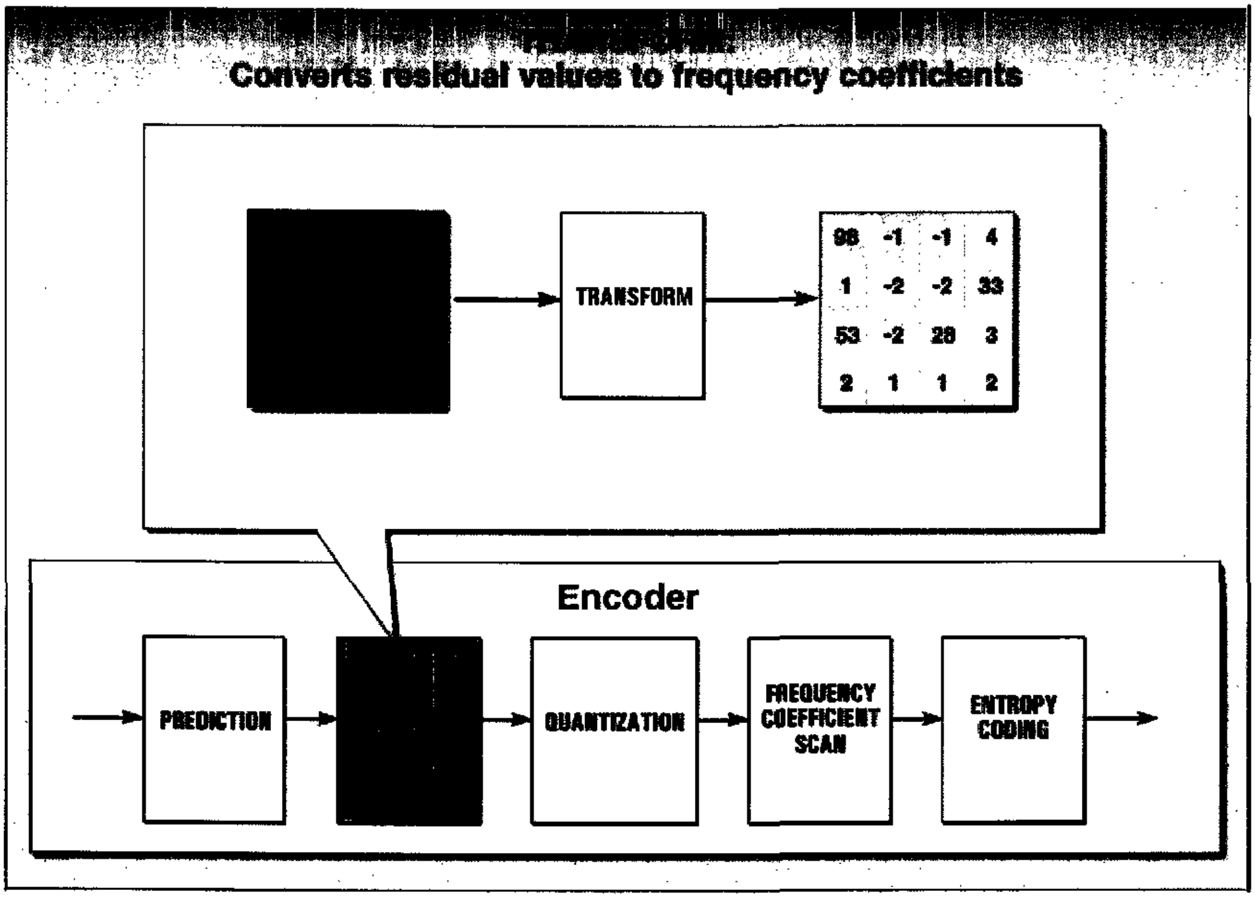
Before going on to the next segment, I would just like to point out that each encoding step makes headway toward eliminating the redundancy in the video signal. We can think of these decoding steps as layers of an onion that we peel back gradually. Of course, one can only peel an onion in one order. So these blocks all fit together in only one order.



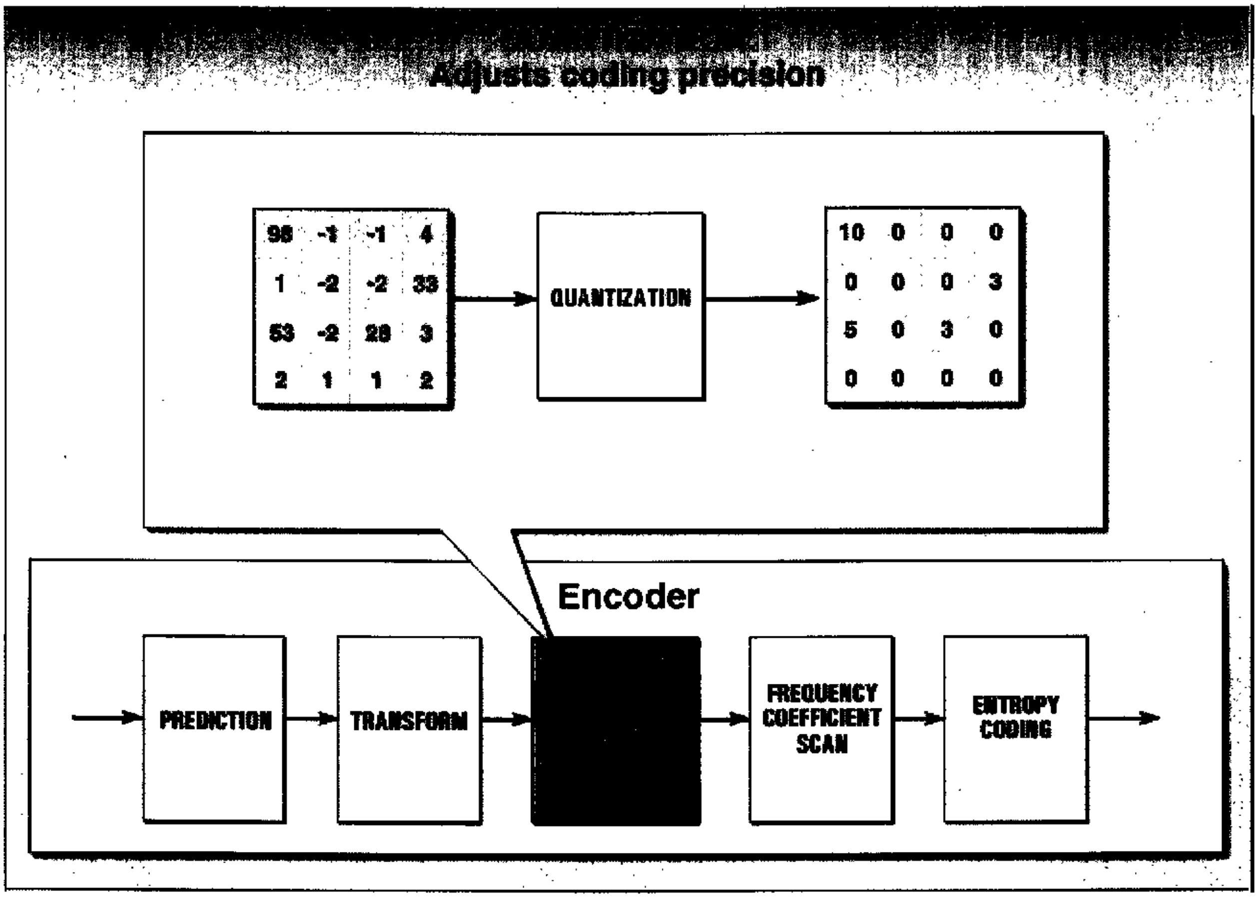
§5,76-79 The first block here is labeled "prediction." The basic idea of prediction is to eliminate redundancy in the video stream from picture to picture in order to reduce the number of bits that need to be transported or stored. So, broadly speaking, if you want to represent a part of the picture, you look for another part of the same picture or a different picture that is very similar. Here is what we are trying to represent, and be found, a so-called predictor, somewhere else that is a lot like it. We compute the difference between those two parts, and that gives us a residual. We process only that. And that is easier to compress.



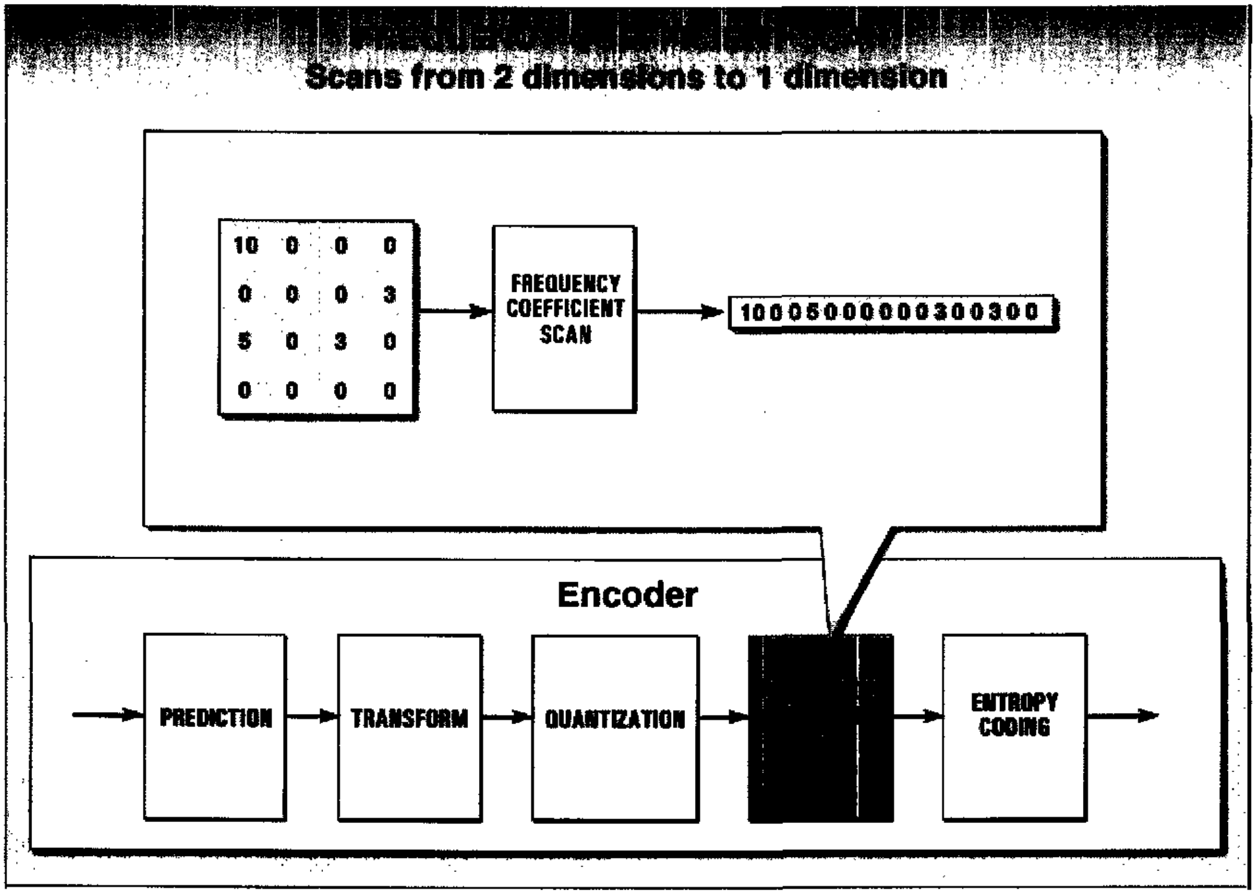
§5,80-82 The second box is labeled "transformer." The basic idea of transformer is to convert pixel values into frequency coefficients. The frequency coefficients are easier to compress than the pixel values themselves. So we are making more headway toward eliminating redundancy.



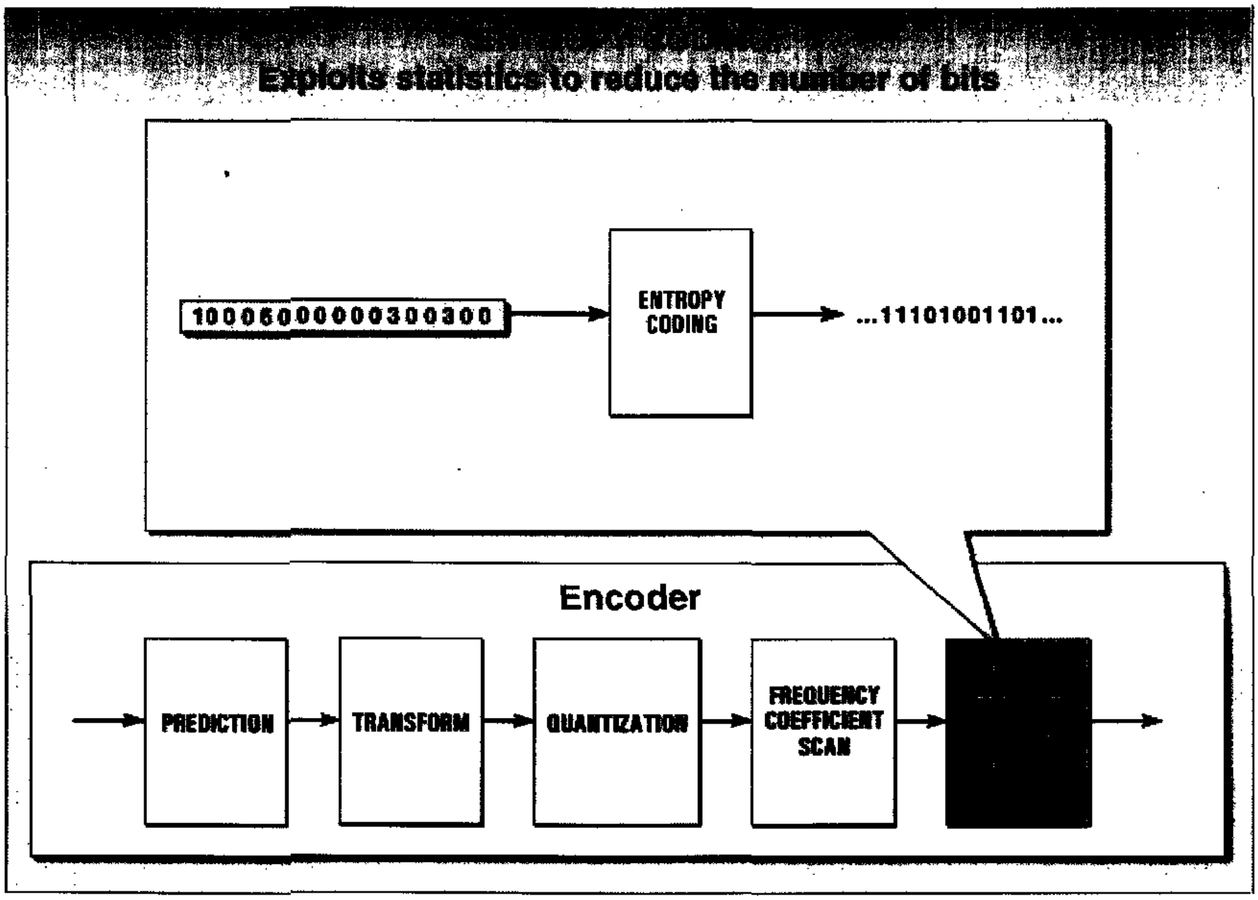
§5,83-85 The third box is labeled "quantization." The basic idea there is to reduce the number of bits required to represent each coefficient. Often certain of the smaller, generally. higher, frequency coefficients will quantize to zero. Quantization is to exploit the fact that the human eye is less sensitive to higher frequency components than to lower frequency components.



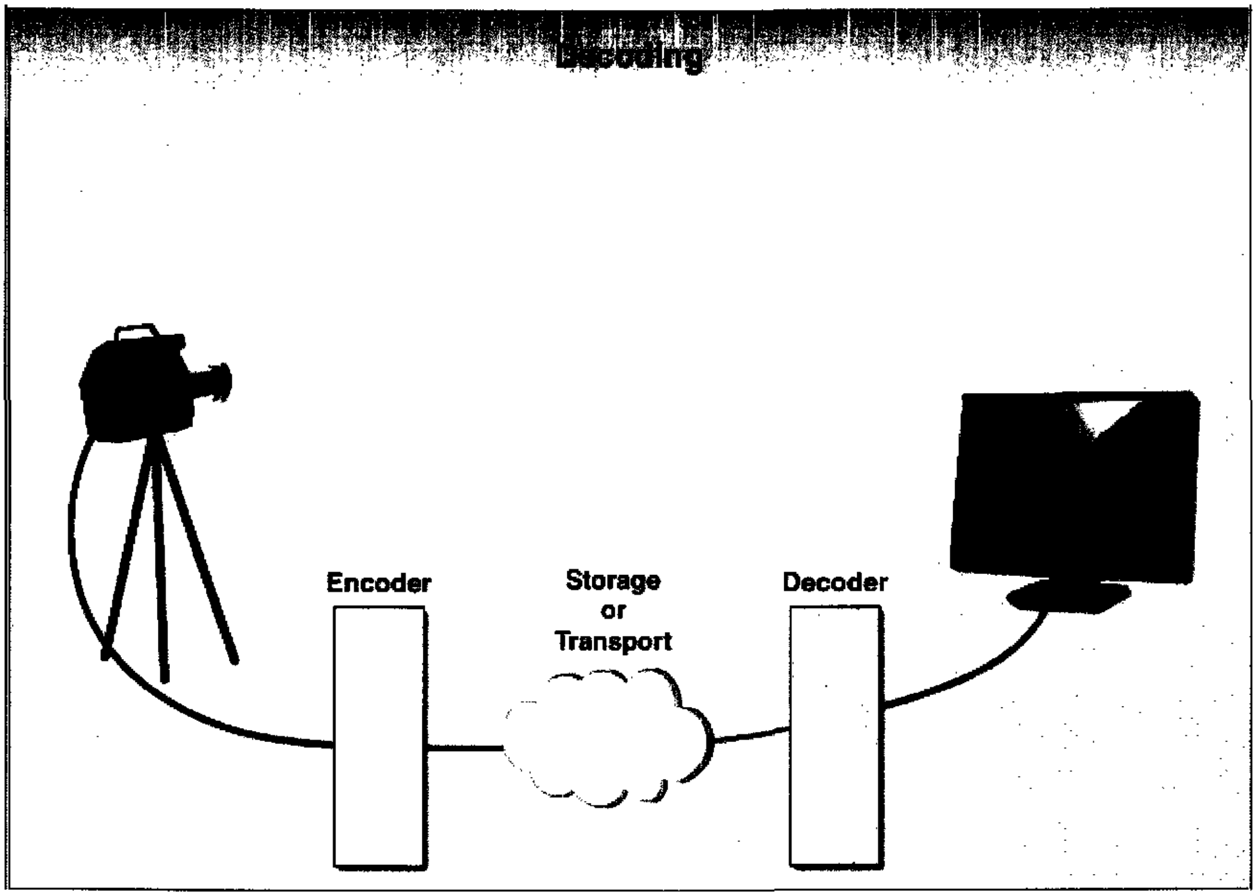
§5,86-88 The fourth box is labeled "frequency coefficient scan." This box scans the frequency coefficients from locations in the two-dimensional array, and reorders them into positions in a one-dimensional array.



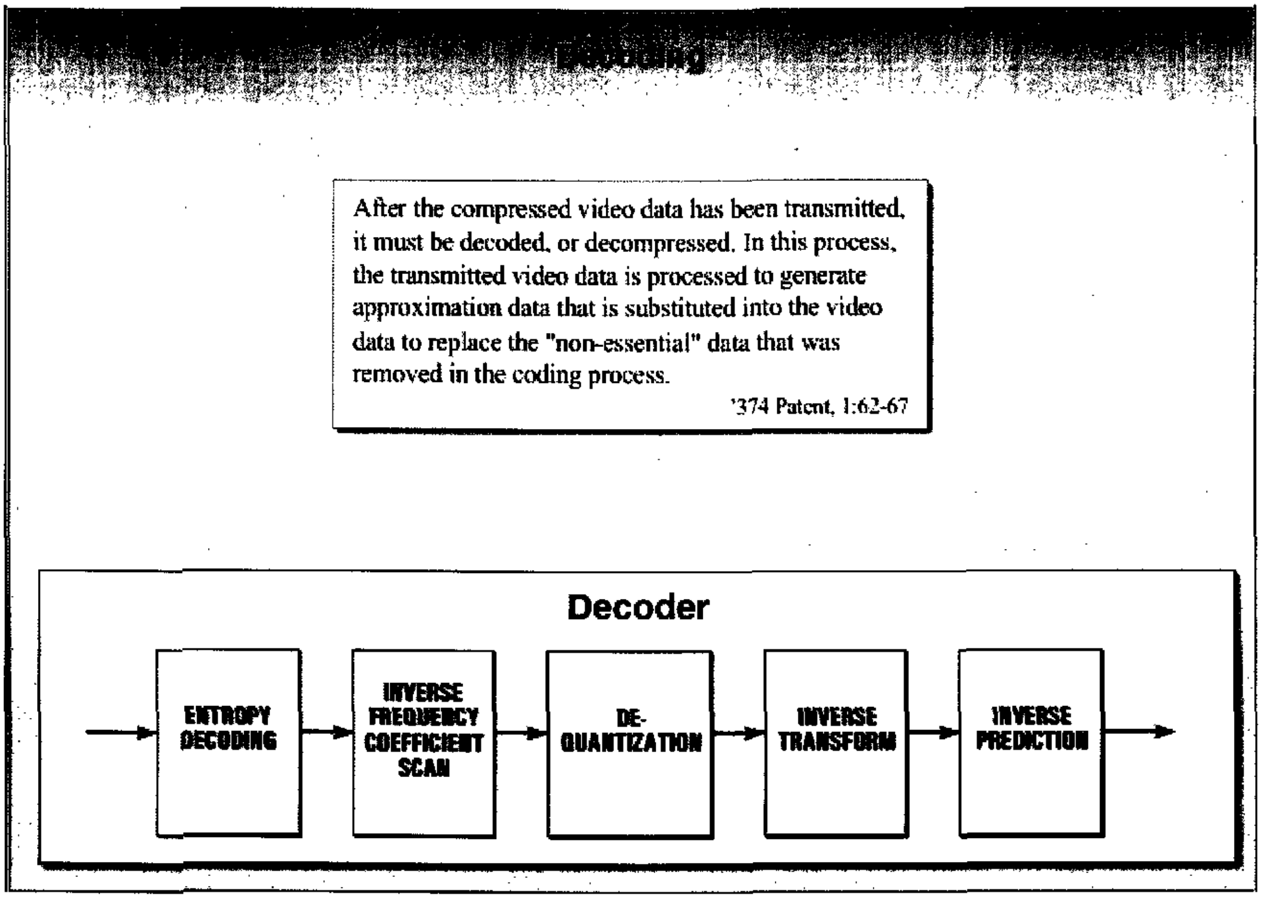
§5,89-92 The fifth box is labeled "entropy coding." The basic idea of entropy coding is to represent the sequence of symbols, a shorter sequence of bits, by exploiting statistics of the information stream. Entropy decoding accepts a one-dimensional stream, a sequence of signals, and that's why we needed the scanning to convert the frequency coefficients into a linear sequence.



§5,93-126 Now I will discuss decoding. As I mentioned, decoding is the reverse of encoding. And if we pause here with the block diagram of the decoder, what we are doing is we are replacing the layers on the onion. Not that we ever do that with an onion, but we do that with video coding. And, of course, we have to replace them in the same order we remove them, but in reverse. So all of these blocks are the inverse operations of the blocks that we discussed in encoding. And those inverses are well understood and well known.

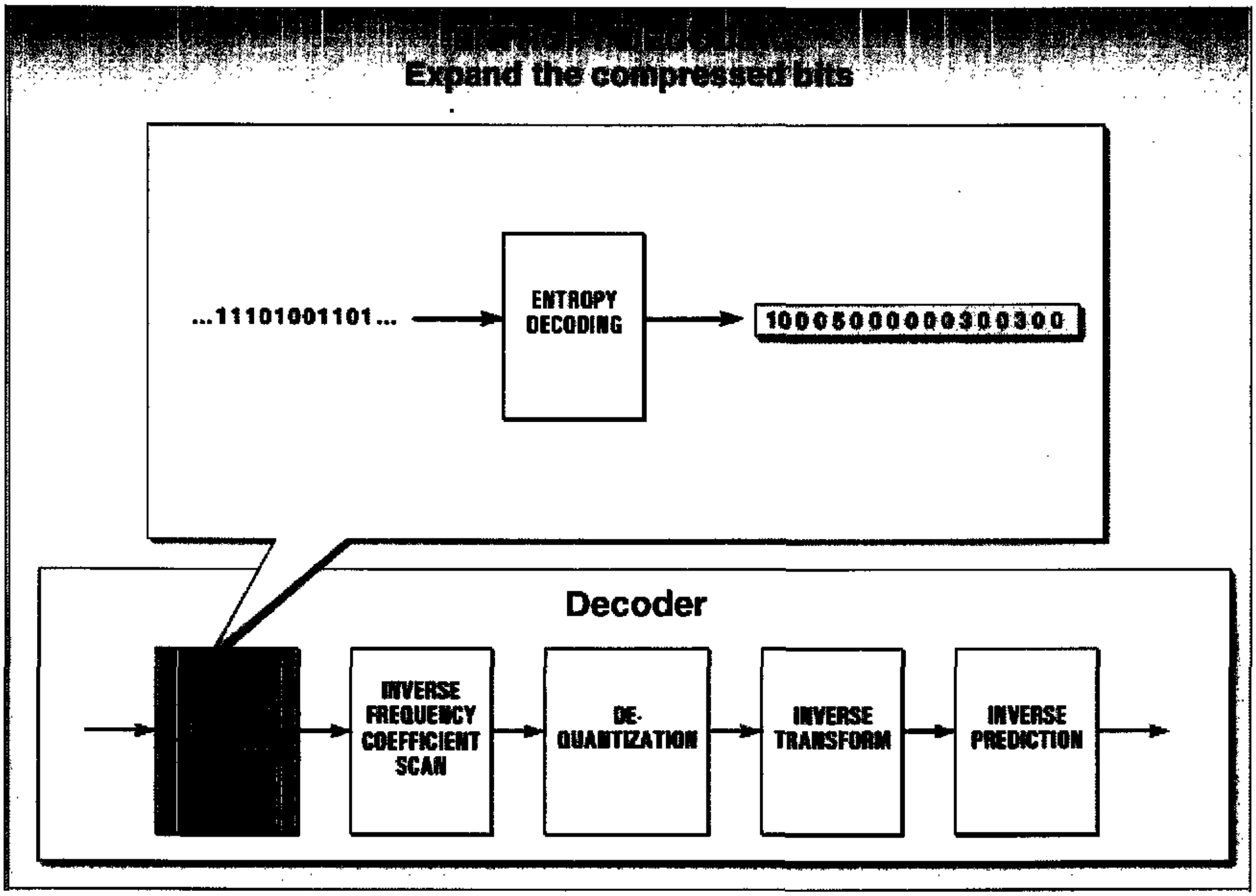


§5,127-200 These boxes, I will just tick through them very quickly.

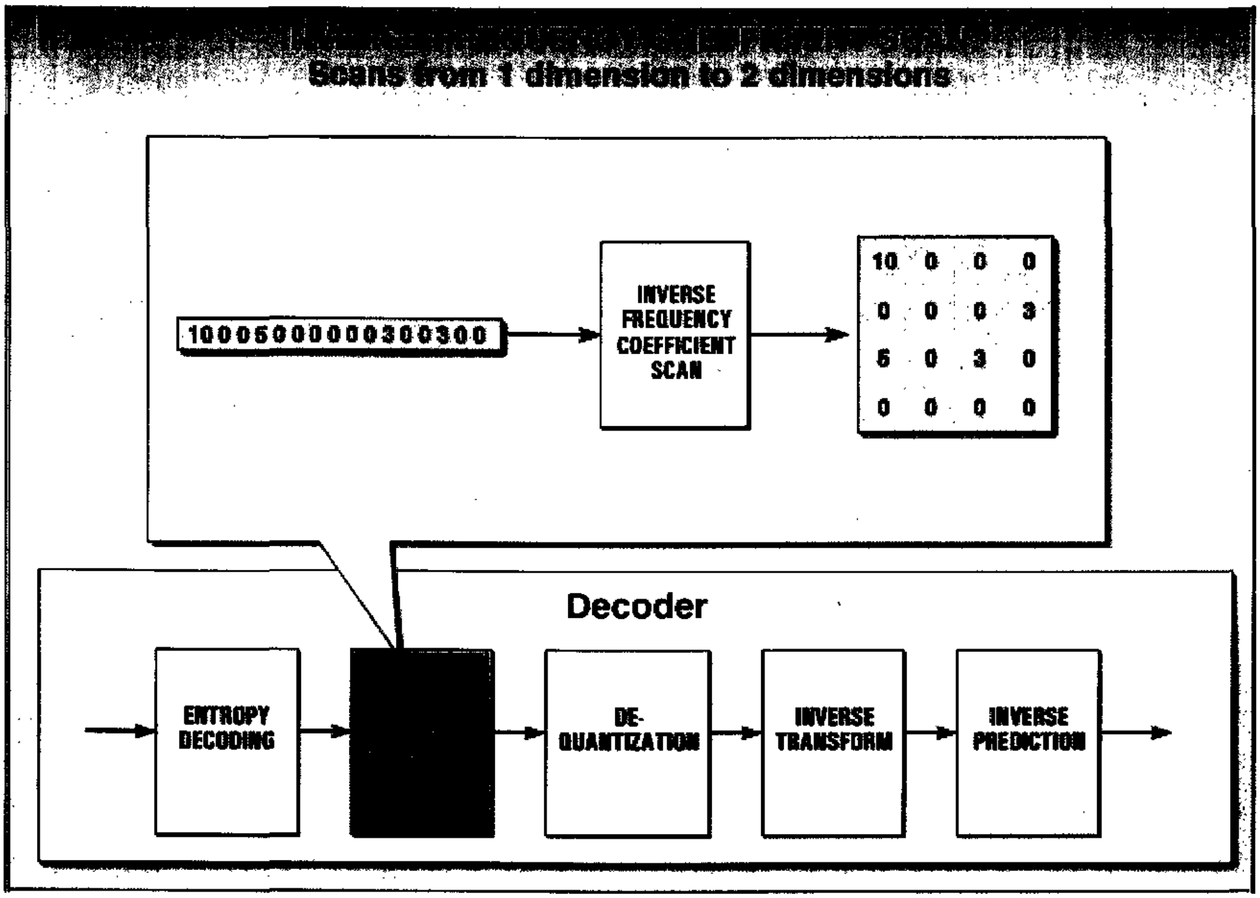


29682505_10

§5,200-204 We have inverse entropy decoding, to take the bitstream and produce the one array of quantized,

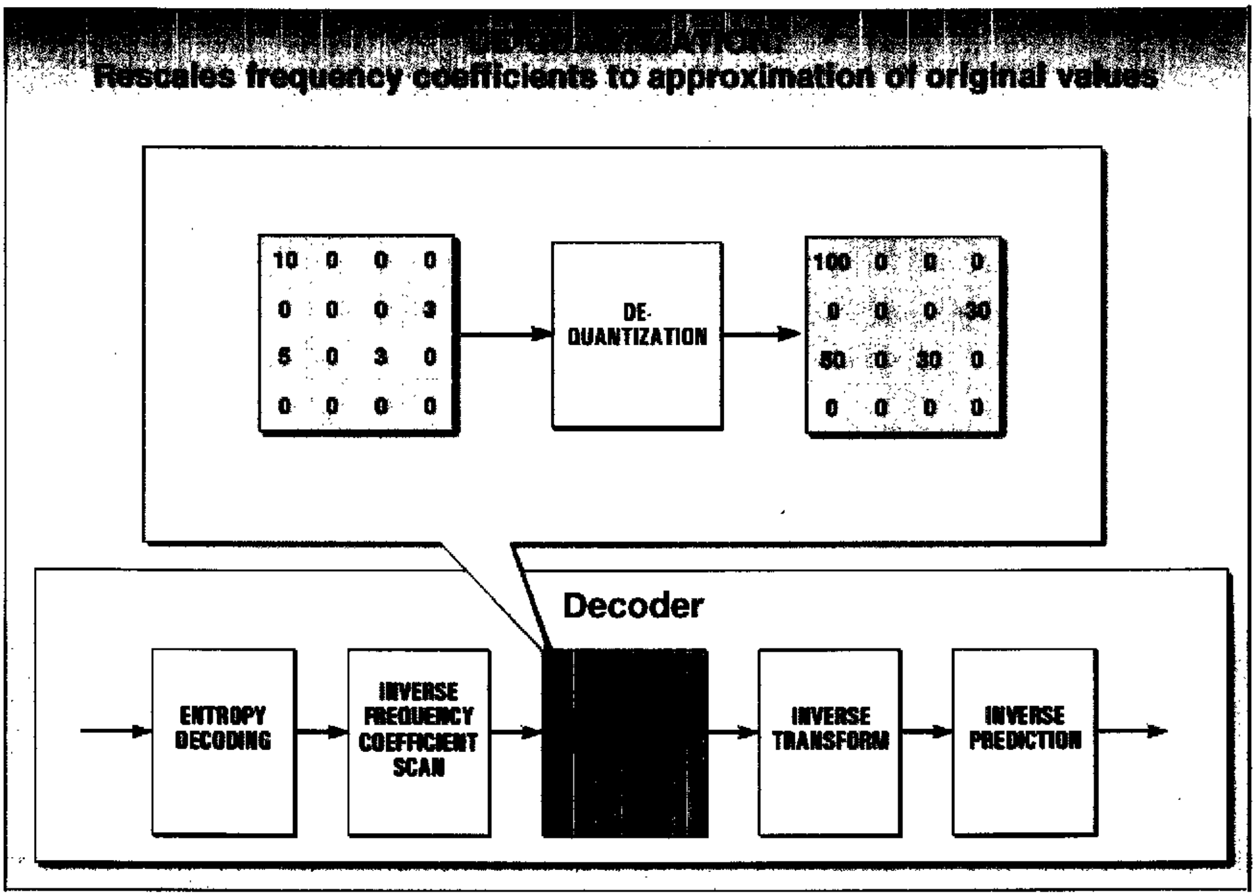


§5,205-207 Then there is the inverse frequency coefficient scan,

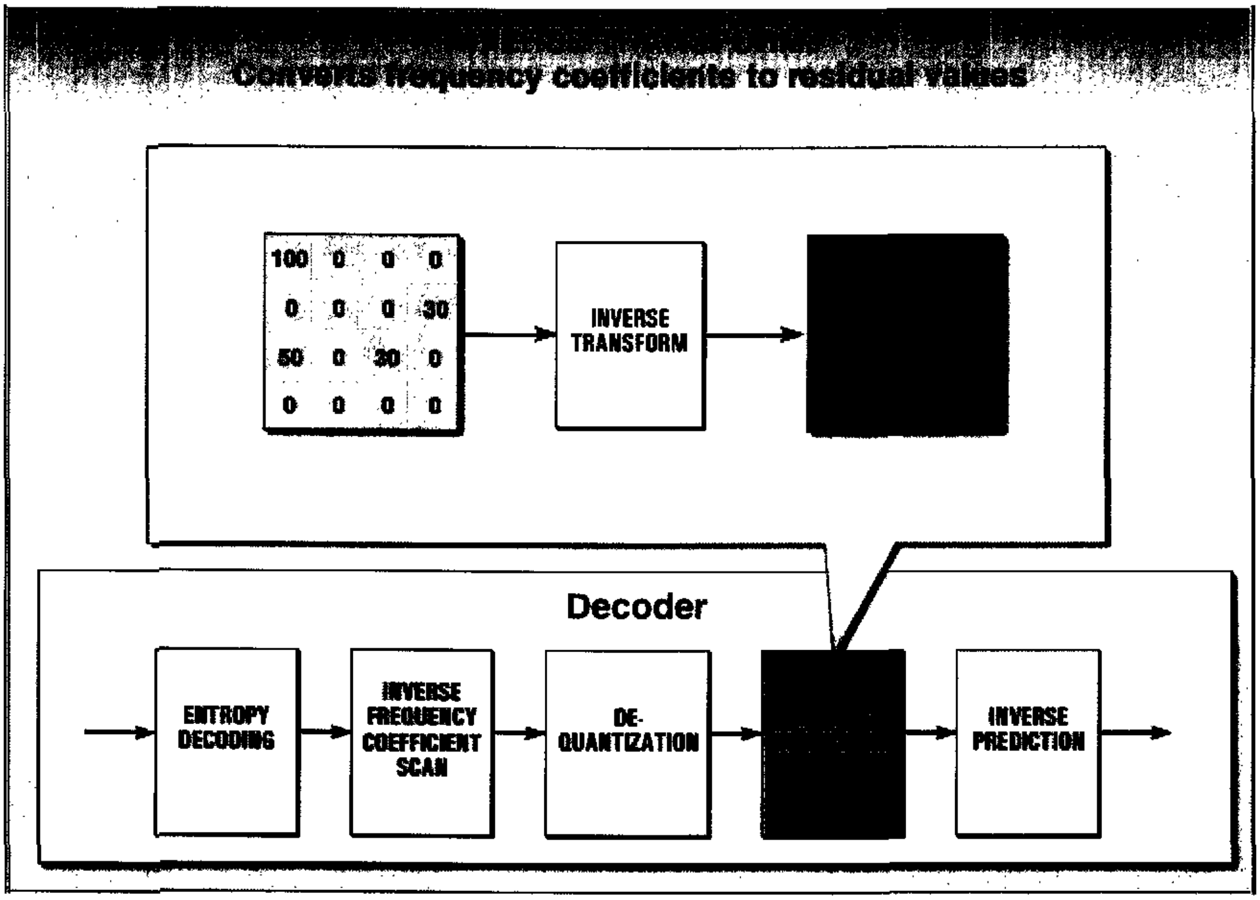


29682505_10

§5,208-210 de-quantization,



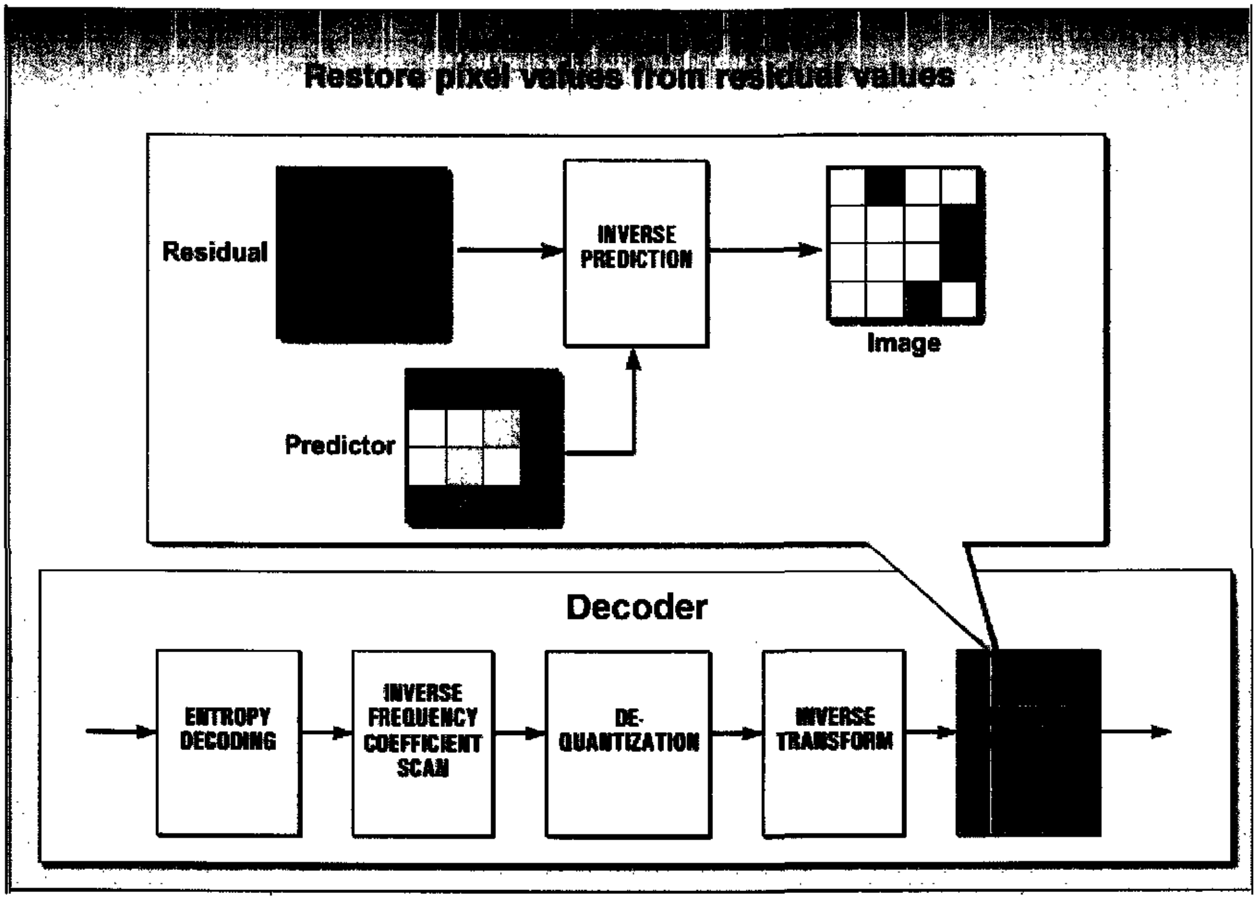
§5,211-213 the inverse transform



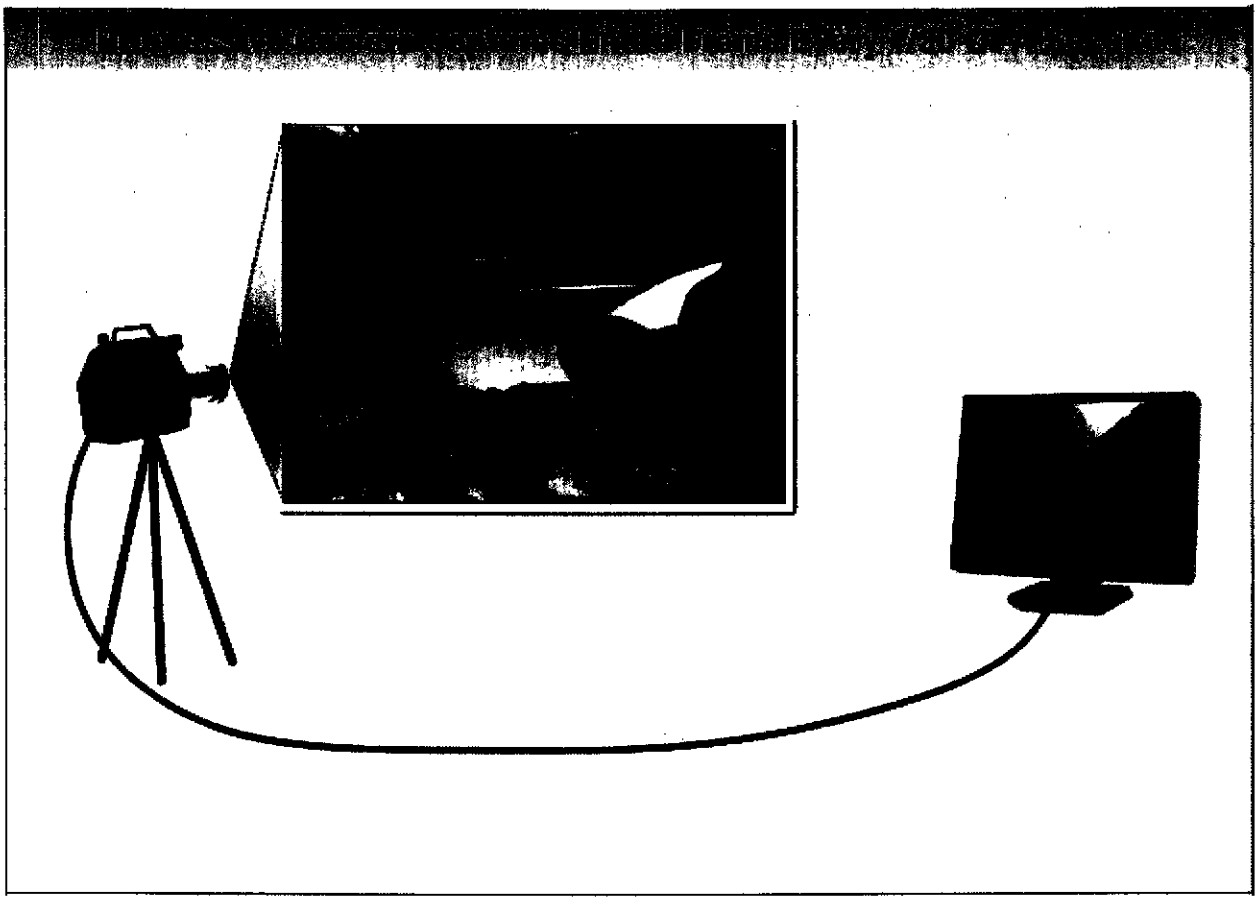
29682505_10

§5,214-216 and inverse prediction.

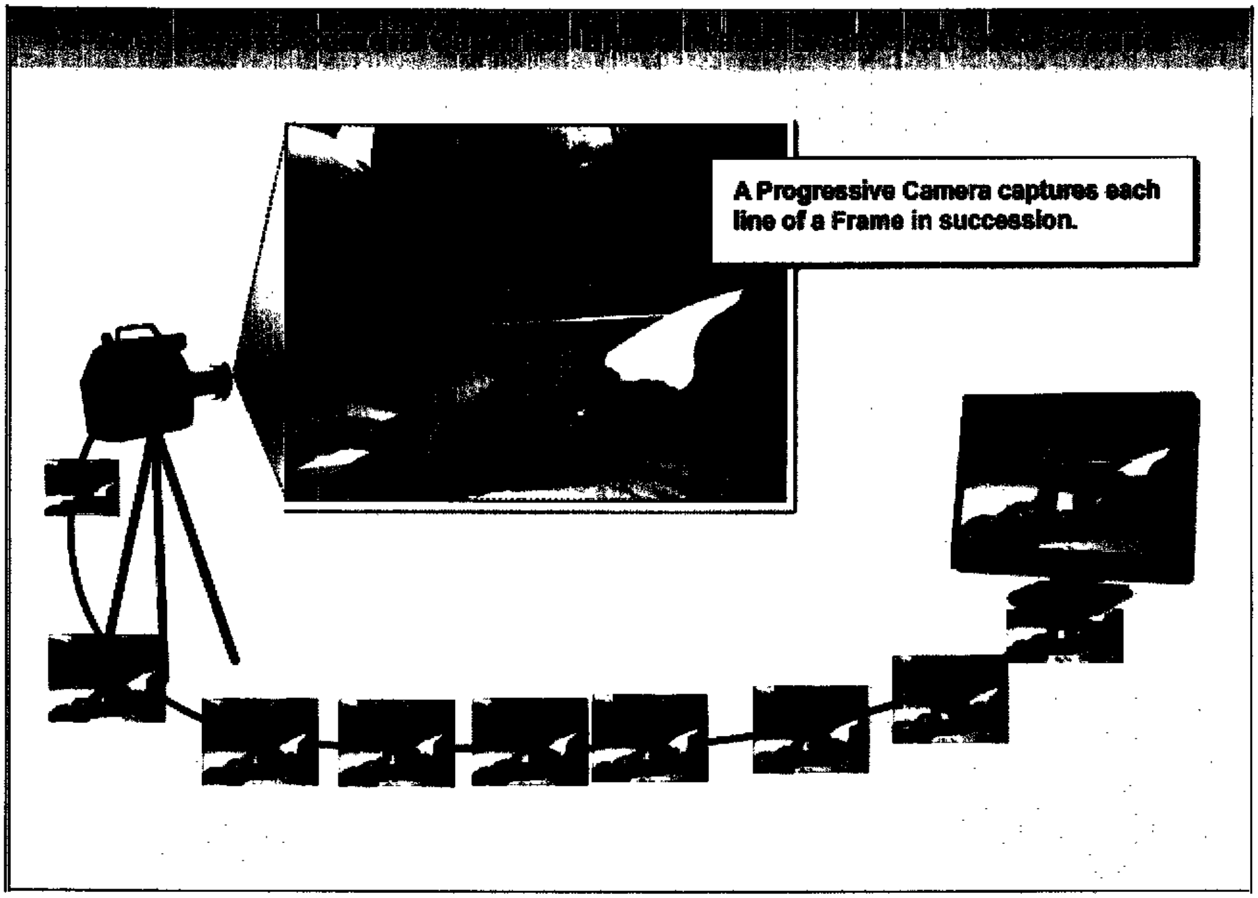
Now, the output of the inverse prediction box is a decoded pixel block. Decoded pixel blocks are used to construct a decoded macroblock. Decoded macroblocks, in turn, are used to construct the current picture for display.



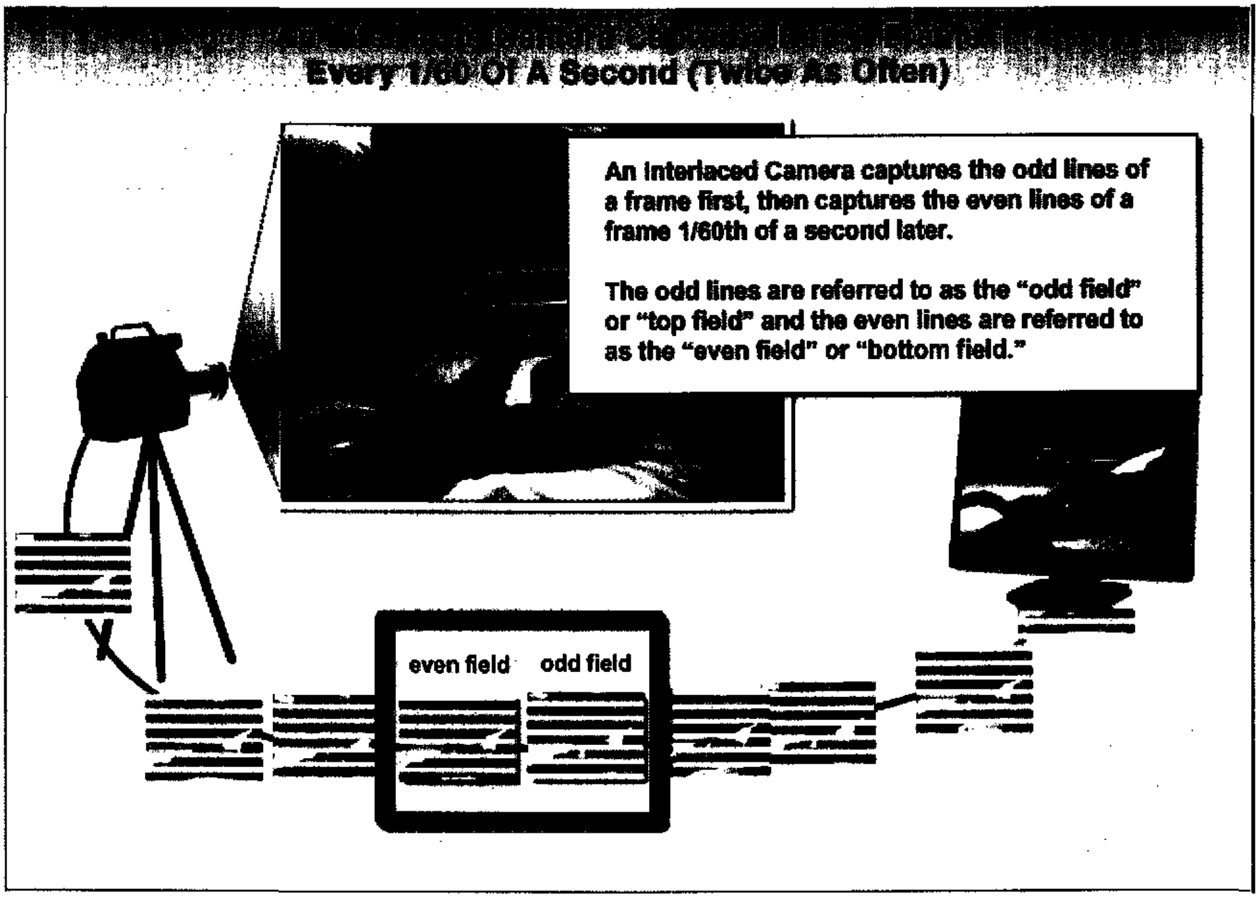
§6,1 So now I have discussed pictures, macroblocks, encoding and decoding. There is one aspect of the background technology that is important, and that is the difference between interlaced video and progressive video.



§6,2-175 For progressive video, every line of a frame is captured by a camera in a single pass from top to bottom. As I illustrate here, in progressive video, an entire frame is sent every 30th of a second.

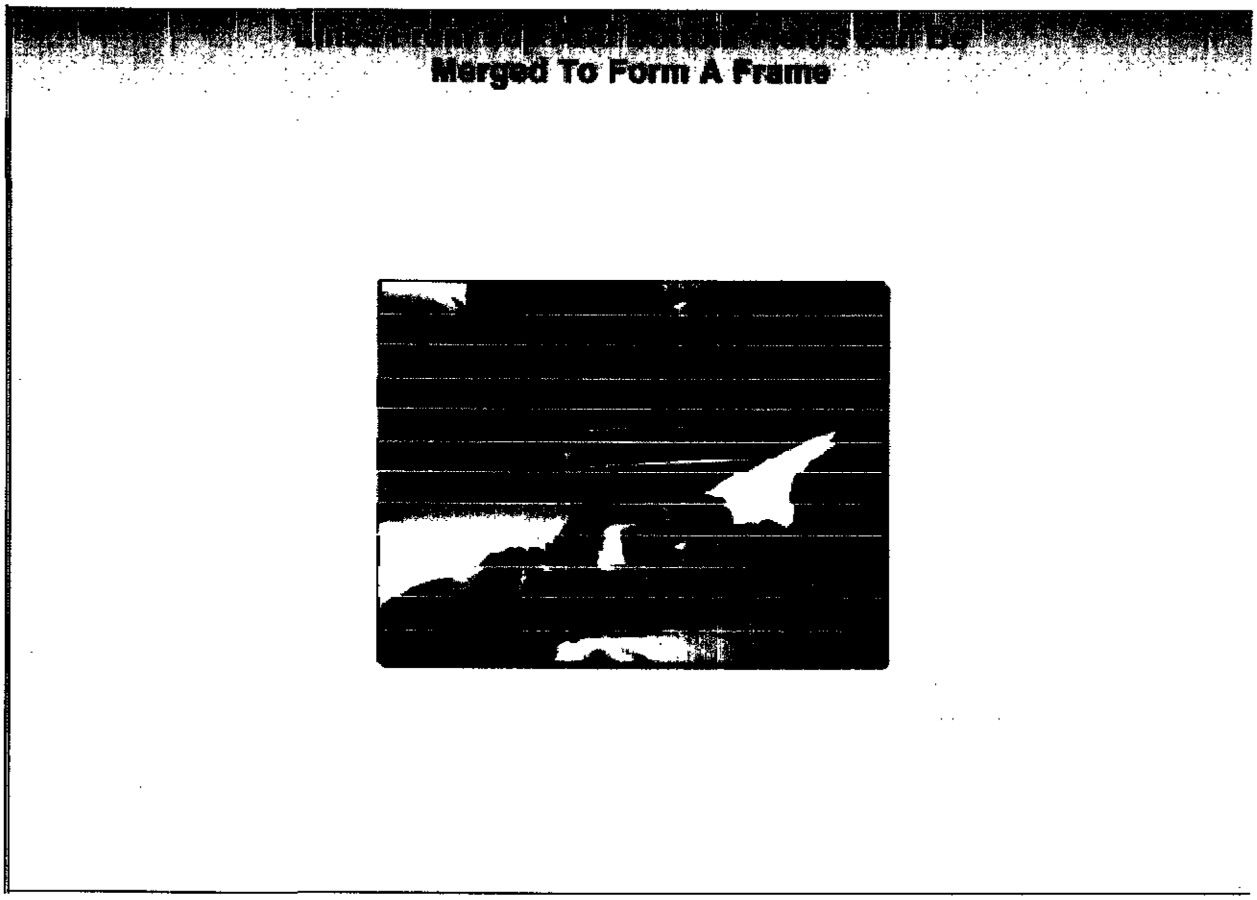


§6,176-332 On the other hand, interlaced video operates by scanning only every other line in each pass. First, the odd scan lines, called the odd or top field, are required. And then the even scan lines, called the even or the bottom field. As I illustrate here, the odd and even fields occur every 1/60th of a second.



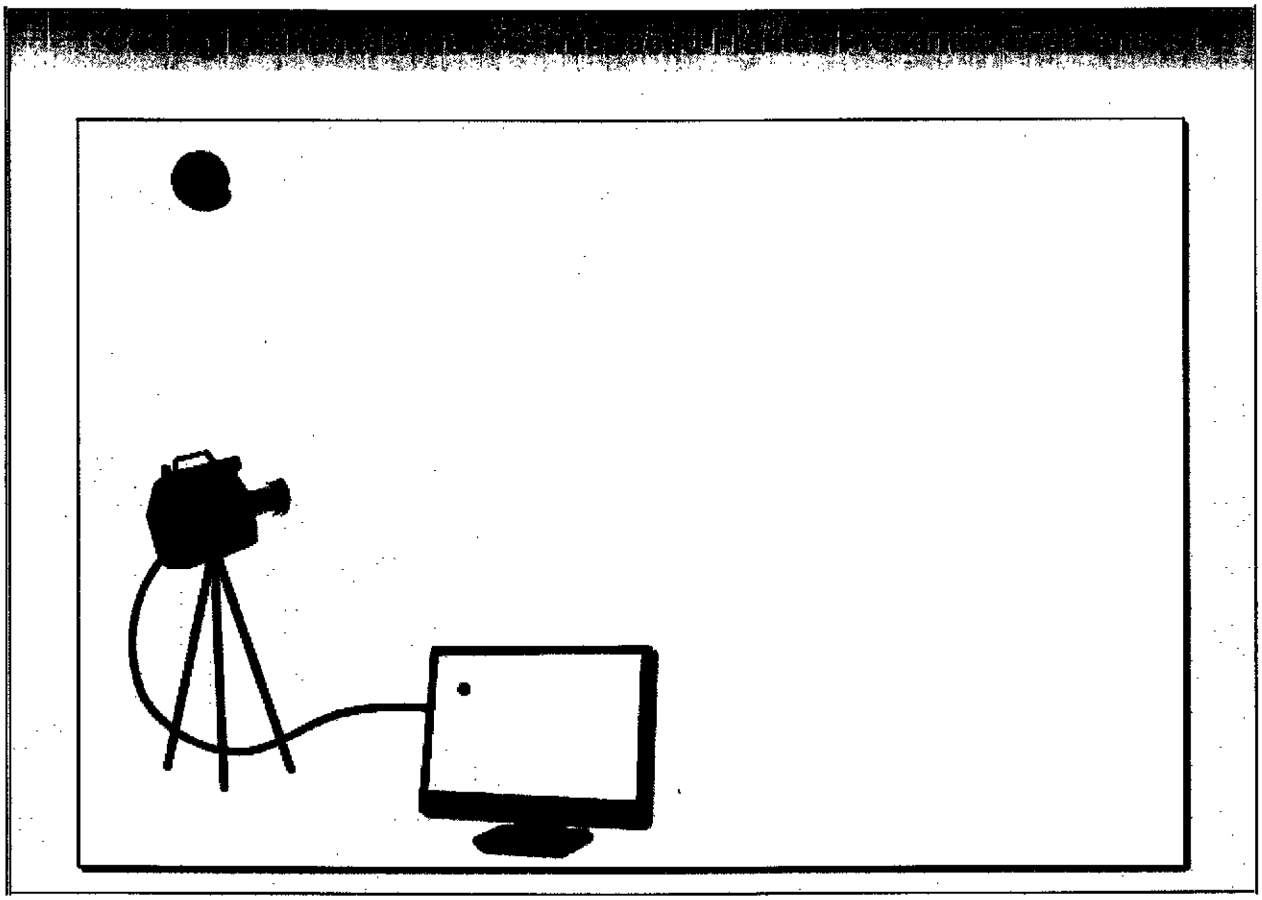
29682505_10

§6,333-499 These two fields are combined to form a picture. We should be combining them shortly.



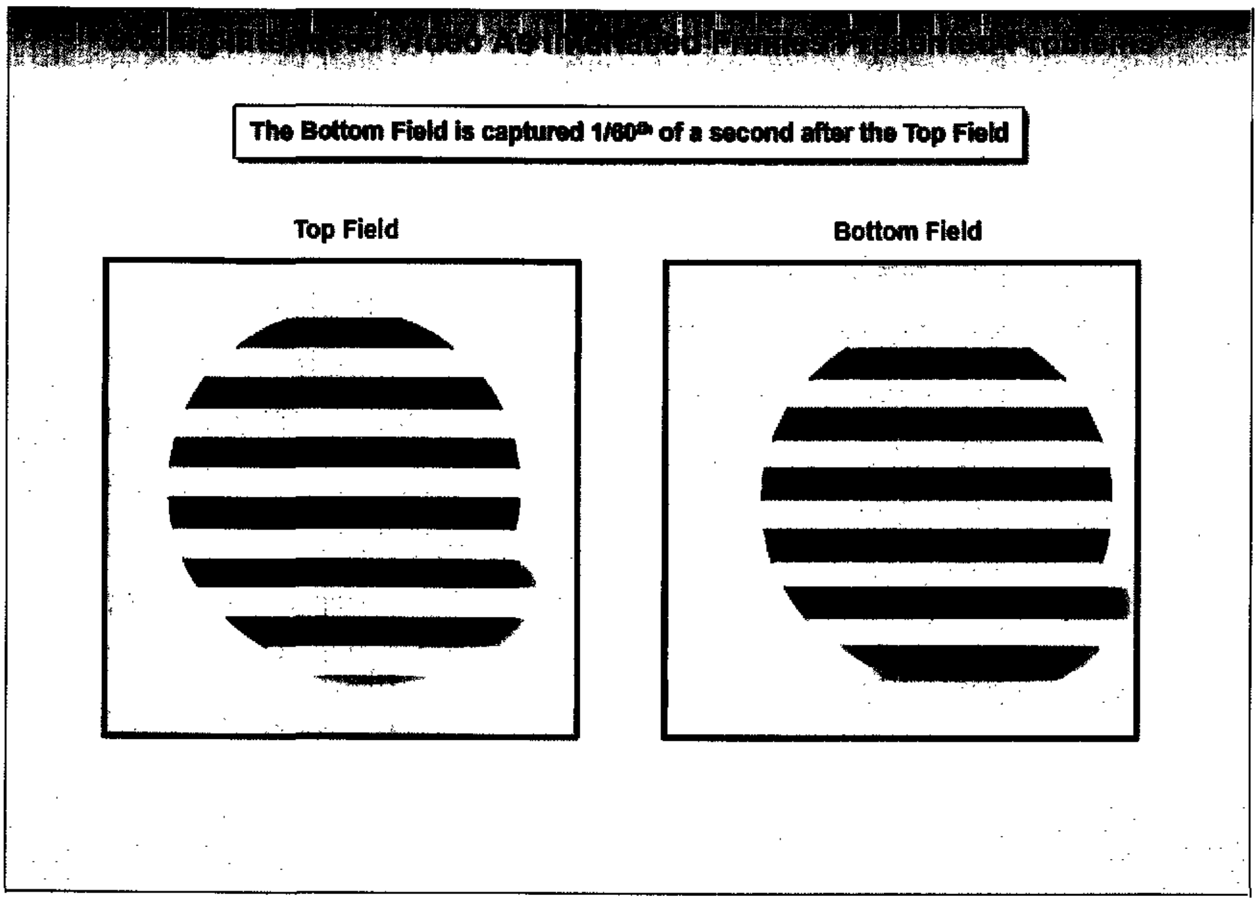
§6,500-569 But coding interlaced video, as frames combine like this, presents problems for images that contain moving objects.

So I will illustrate this problem using an example of a moving red ball.

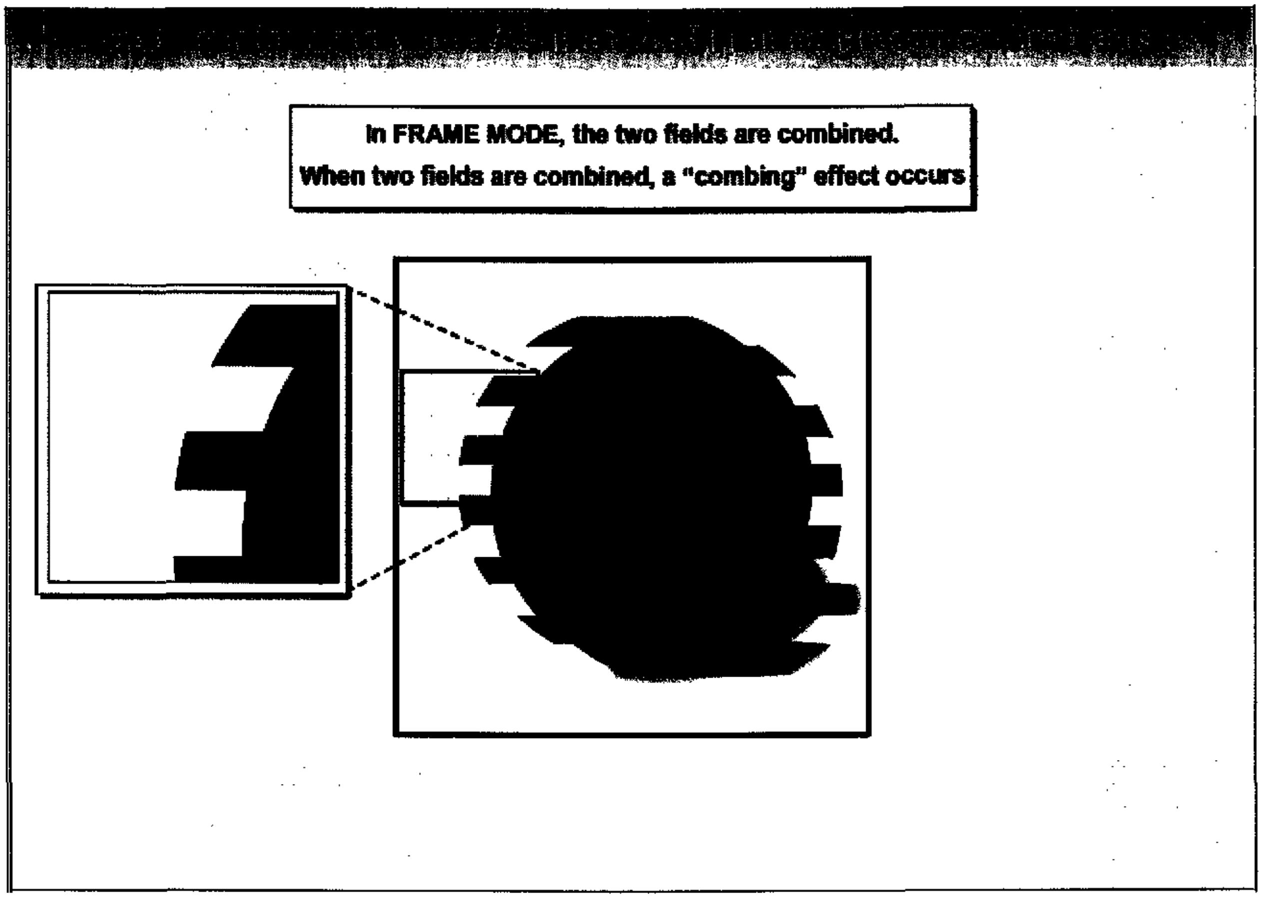


§6,571-731 Here we see a video camera recording the red ball as it moves across a still background. When the camera captures video in an interlaced form, it captures half of the scan lines, the top field in the first pass, and the other half of the scan lines, the bottom field, in the second pass, a 60th of a second later.

In this example the box around the ball is a macroblock boundary. The top field on the macroblock on the left is captured a 60th of a second before the bottom field of the macroblock on the right, so the ball is slightly further to the right in the field on the right.

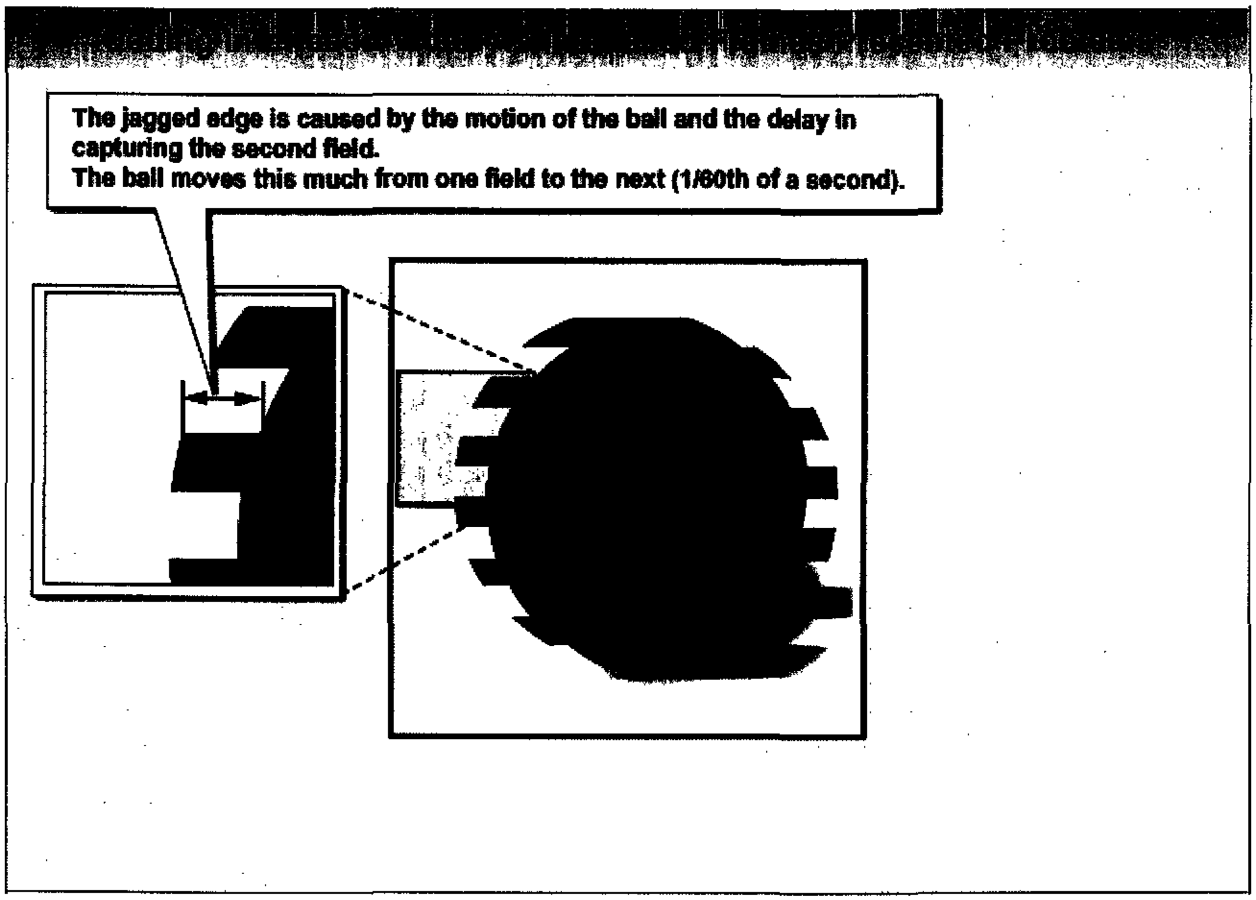


§6,732-785 If we combine these two fields of the macroblock directly, you can see that there is a combing effect around the edges of an object in motion, which I have shown here in the inset.

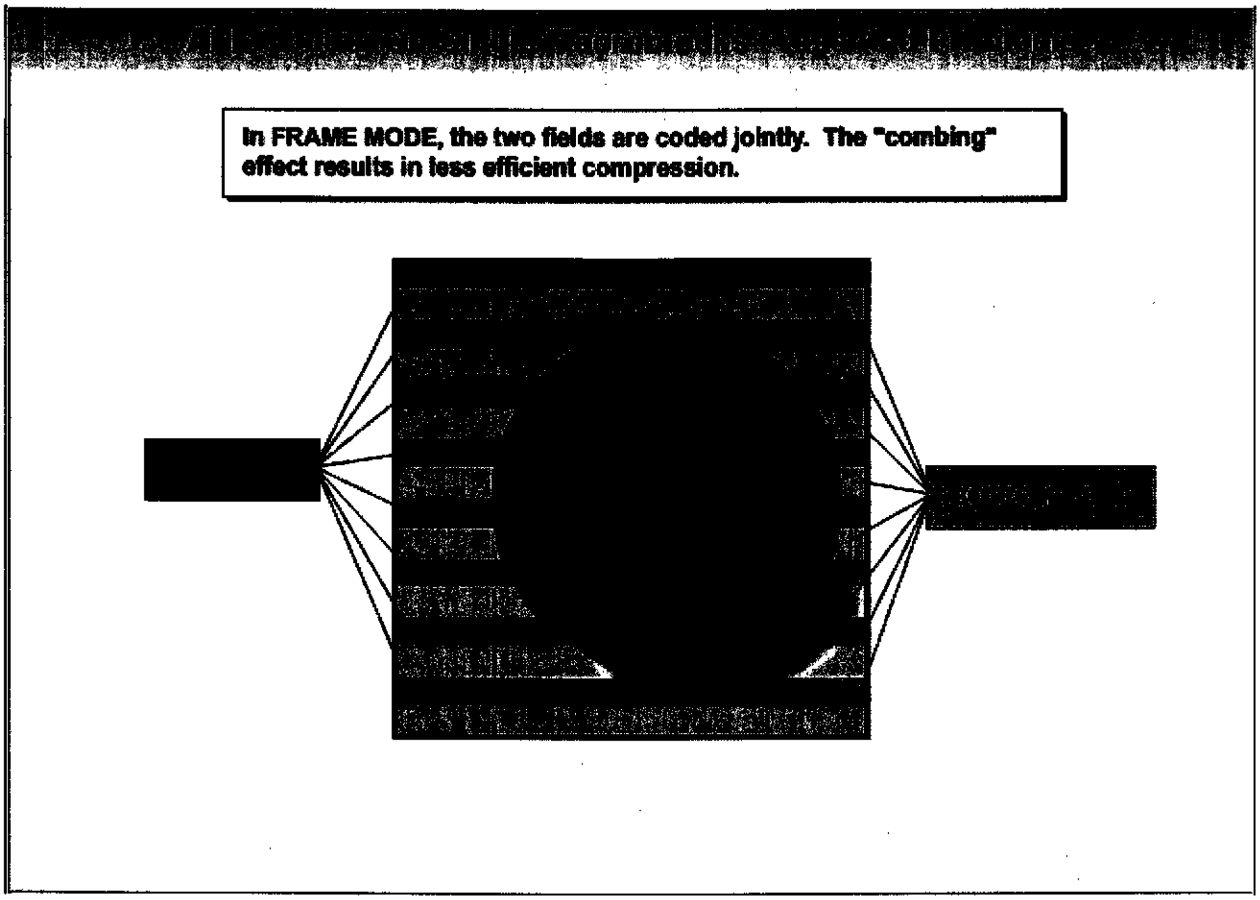


§6,786 By zooming in on the edge of the ball, you can see how much it moved between the time that the first field was captured and the time that the second field was captured.

In prior art encoding, this single macroblock could be encoded in one of two modes, in frame mode or in field mode. This is called adapted frame/field coding, or AFF.

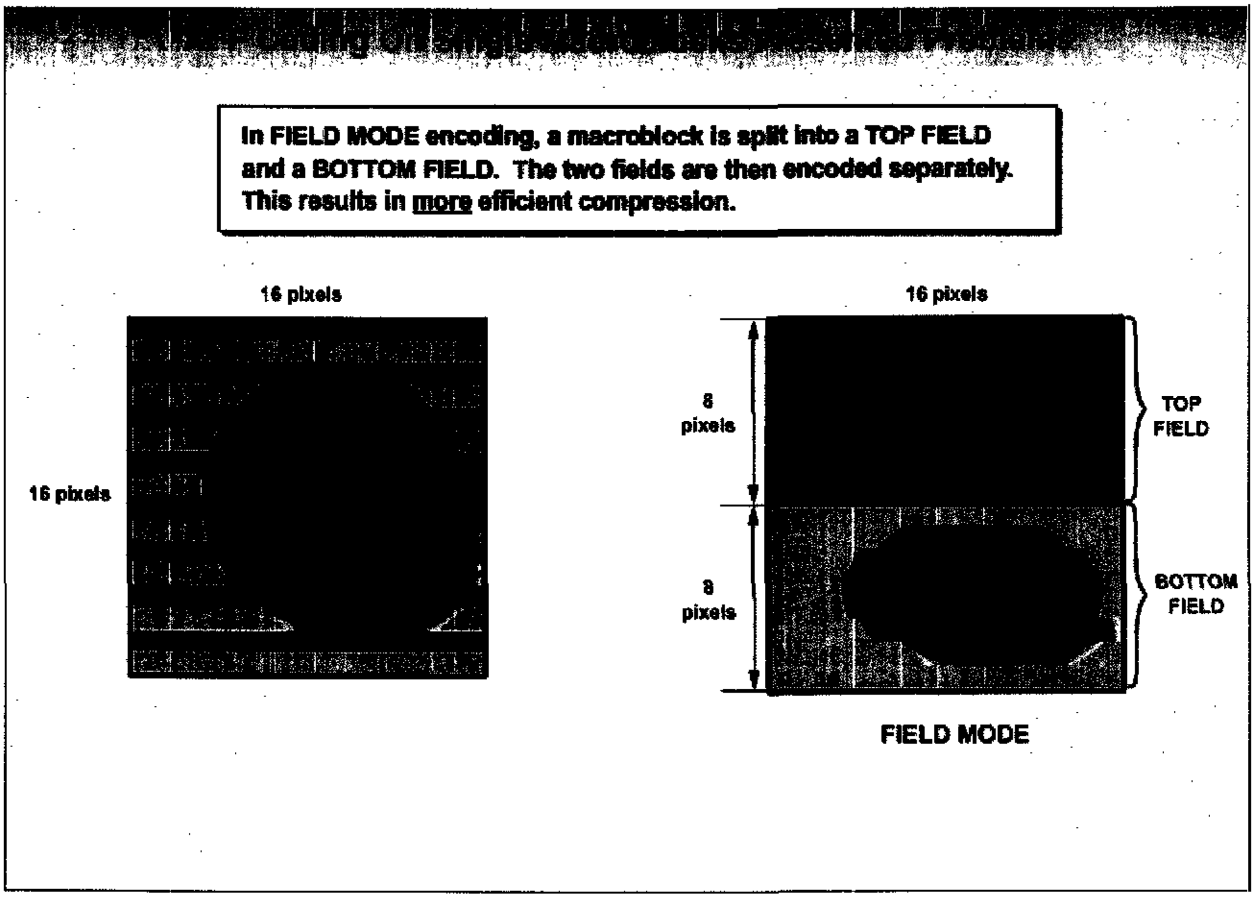


§6,787 In the frame mode, the two fields of the macroblock were encoded jointly. The combing effect caused by combining the two fields of the moving image resulted in less efficient compression.

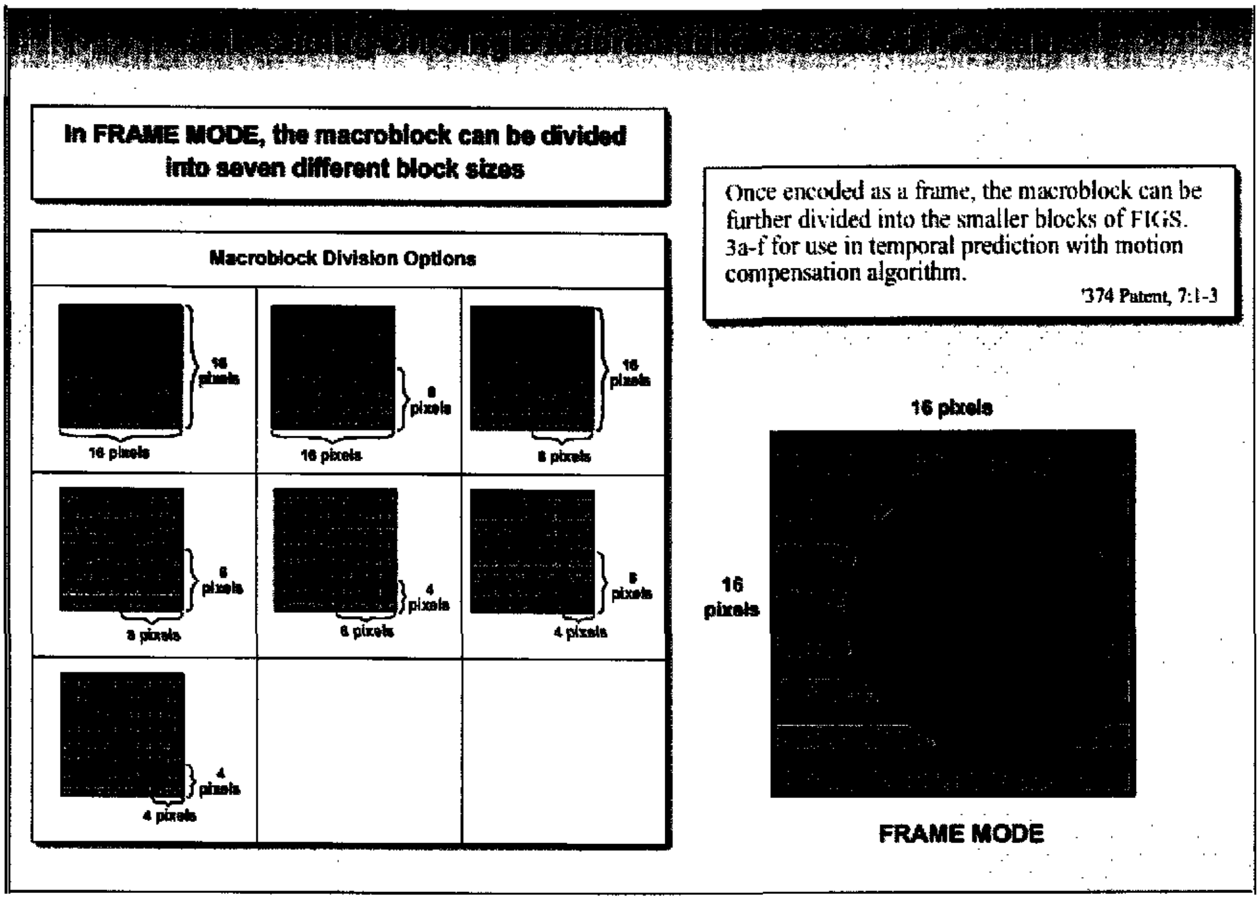


29682505_10

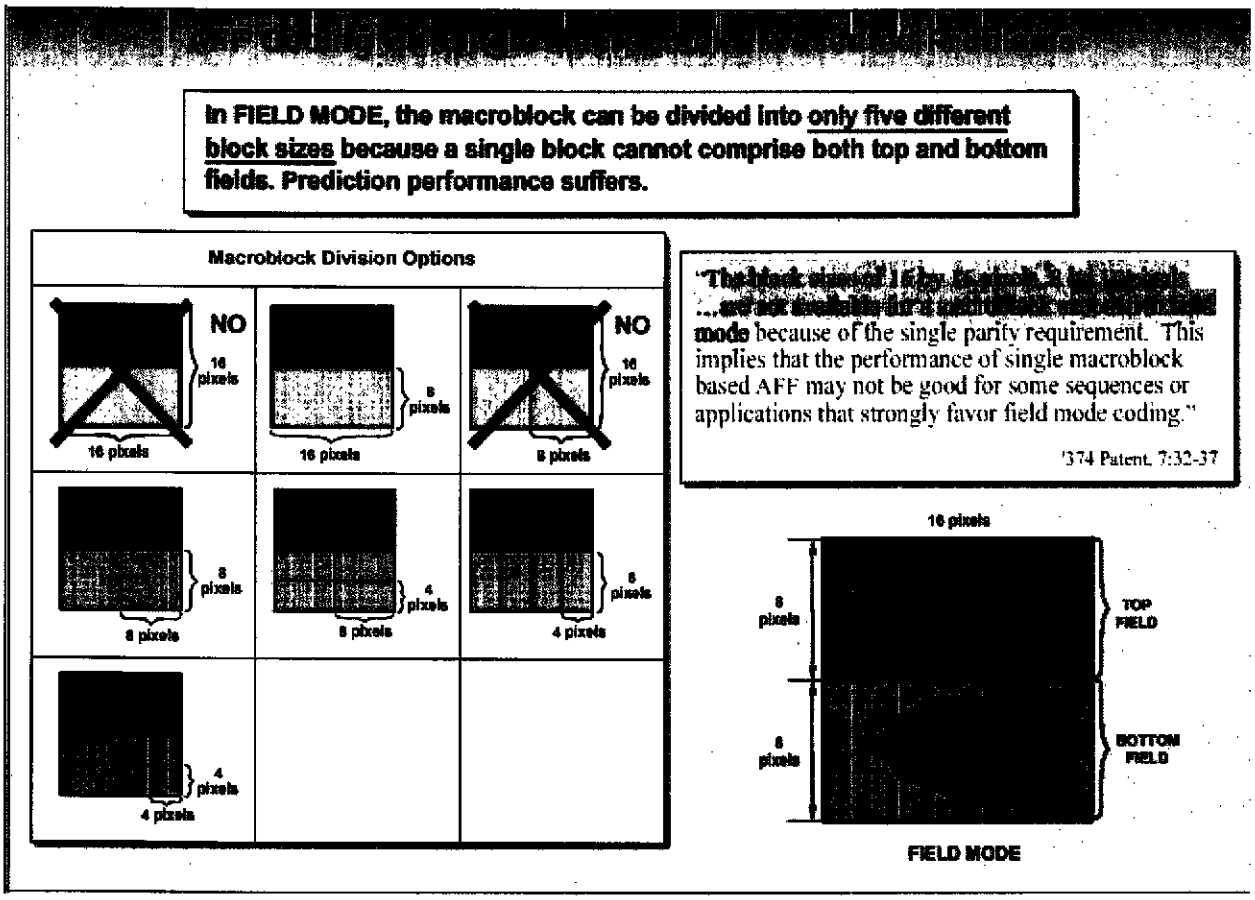
§6,788-1066 In field mode, the two fields were encoded separately. The macroblock was split into two eight-line groups, a top field group and a bottom field group. By eliminating the combing effect, such a macroblock could be compressed more efficiently.



§6,1067-1110 Now, for prediction, macroblocks could be divided into smaller-size blocks. In the frame mode, the 16 by 16 macroblock could be divided seven different ways, as depicted on the figure on the left.



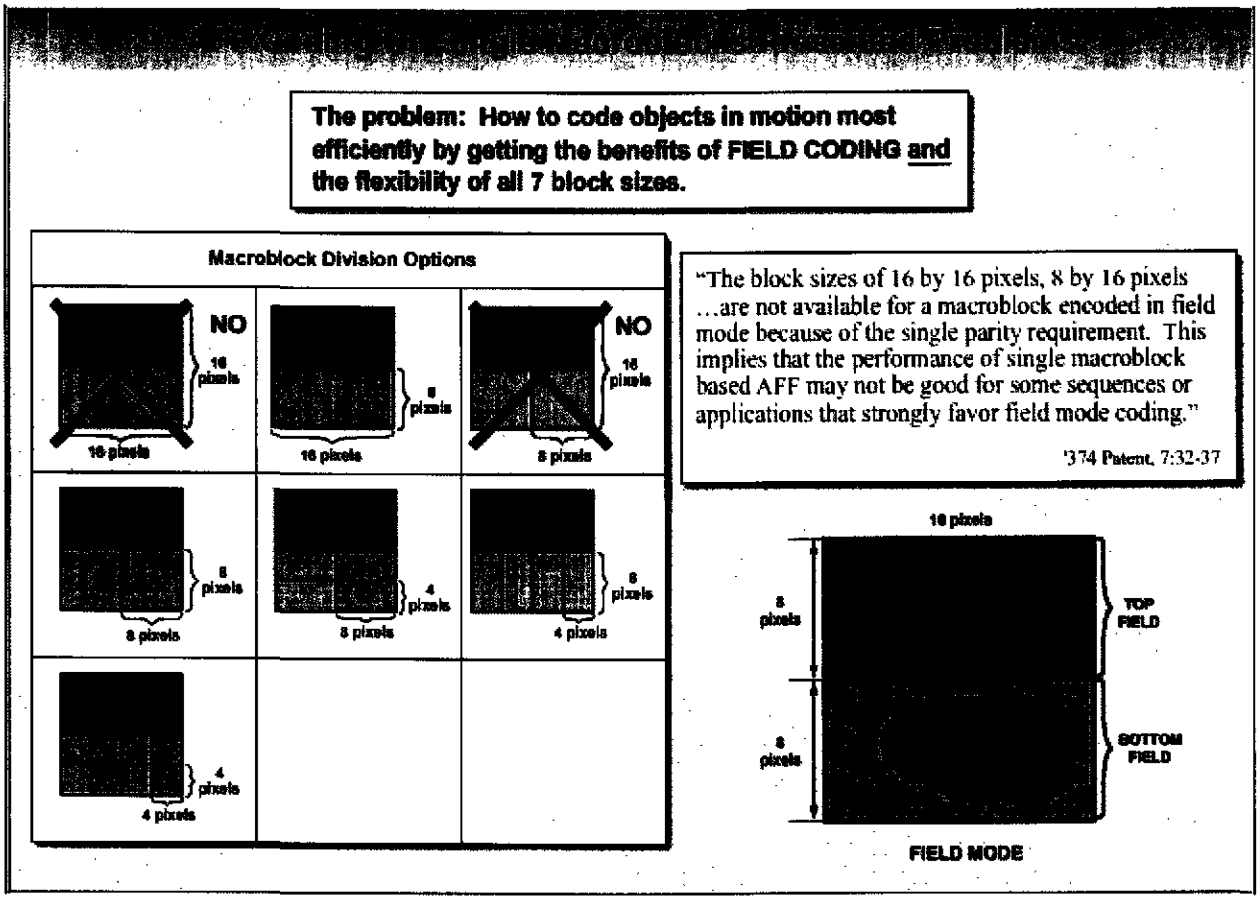
§6,1111 In the field mode, because the macroblock was first divided into a 16 by 8 top field, and a 16 by 8 bottom field, the macroblock could only be divided five different ways, as depicted in the figure on the left. In particular, the block sizes of 16 by 16 and 8 by 16 were not available, because in field mode a block cannot contain lines from both the top field and the bottom fields. This is called the single parity requirement.



29682505_10

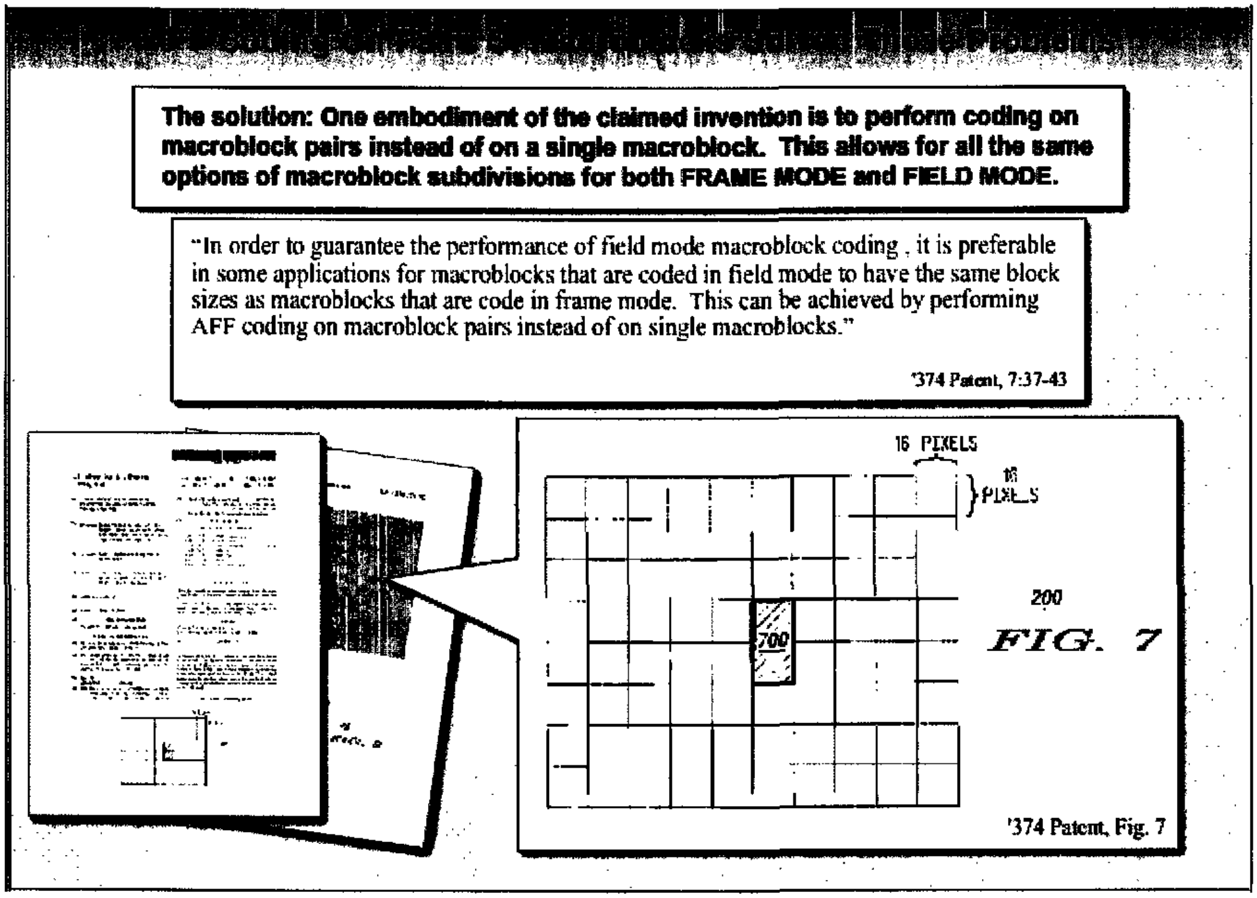
§6,1112 So this was a basic problem with coding techniques prior to the invention. A macroblock in field mode could not be divided the same seven ways as the frame mode macroblock, because when all seven block sizes were not available in field mode, the prediction performance suffered.

The patent states, "This implies that the performance of single macroblock-based AFF may not be good for some sequences or applications that strongly favor field coding.



29682505_10

§6,1113-1121 The solution arrived at by the Motorola inventors was to use multiple neighboring macroblocks instead of a single macroblock for adaptive frame/field coding. For example, Figure 7 of the patent shows a vertical pair of macroblocks that can be used in adaptive frame/field coding.



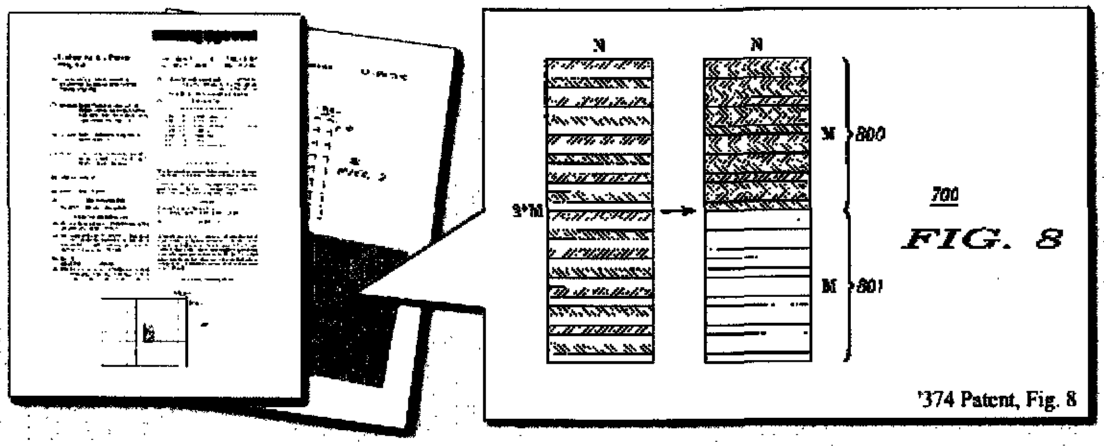
29682505_10

§6,1122-1125 Figure 8 shows how the even and odd lines of a vertical pair of macroblocks can be split into a top macroblock with only odd lines and the bottom macroblock with only even lines.

The solution: One embodiment of the claimed invention is to perform coding on macroblock pairs instead of on a single macroblock. This allows for all the same options of macroblock subdivisions for both FRAME MODE and FIELD MODE.

"In order to guarantee the performance of field mode macroblock coding , it is preferable in some applications for macroblocks that are coded in field mode to have the same block sizes as macroblocks that are code in frame mode. This can be achieved by performing AFF coding on macroblock pairs instead of on single macroblocks."

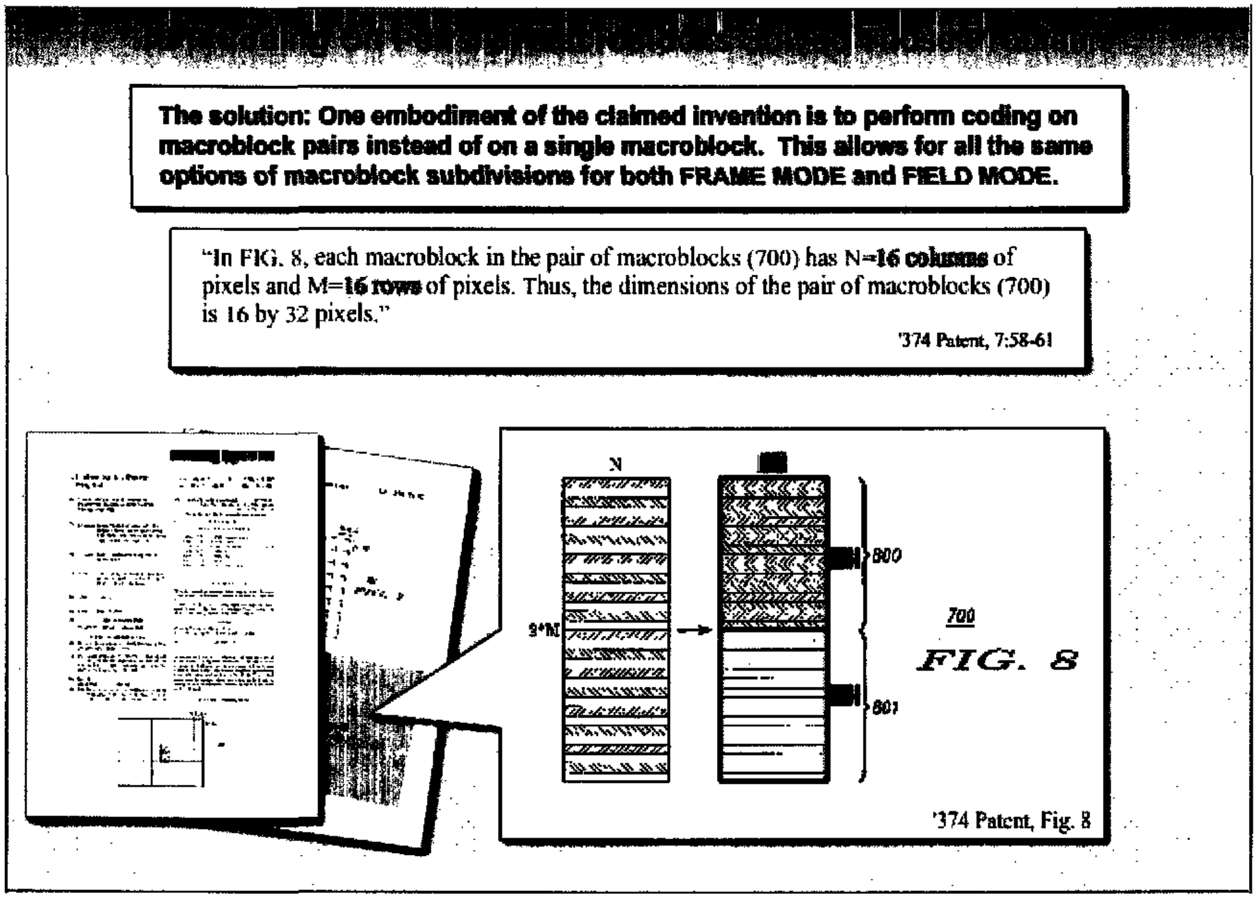
'374 Patent, 7:37-43



'374 Patent, Fig. 8

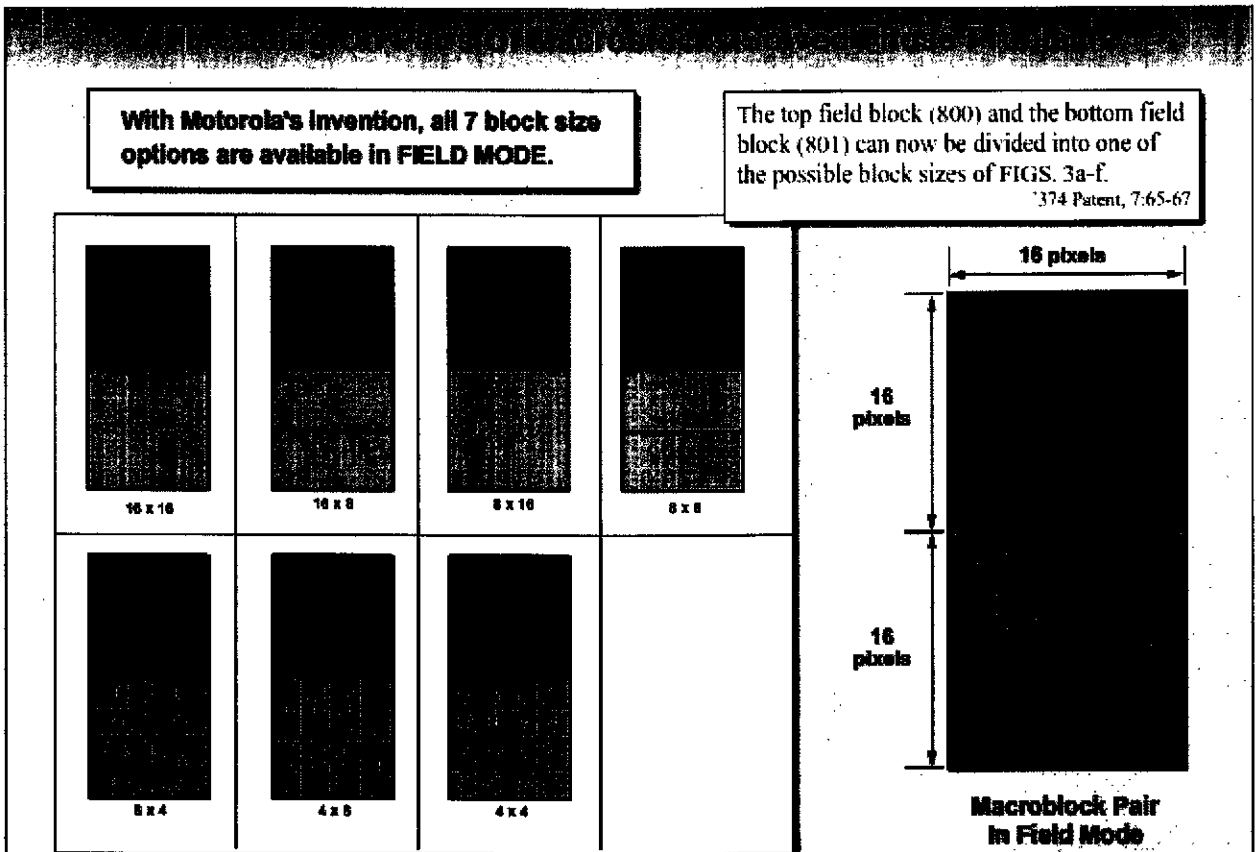
29682505_10

§6,1126 The specification describes Figure 8 as a pair of macroblocks, each of which has 16 columns of pixels and 16 rows of pixels.



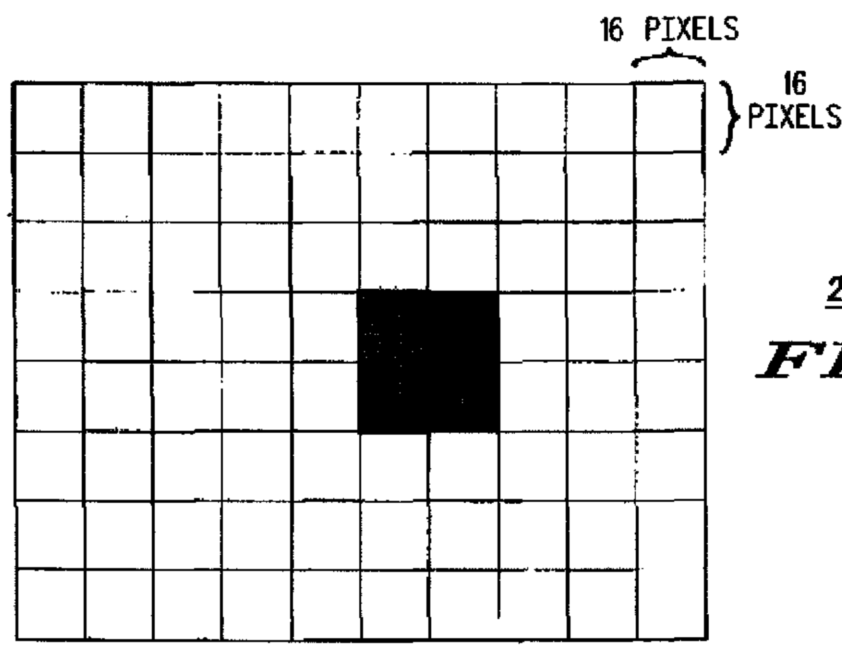
29682505_10

§6,1127-1408 Now, because the top field macroblock and the bottom field macroblock are now both 16 by 16, each macroblock can be divided the same seven ways as a 16 by 16 frame macroblock. We get back this, and we get back this. This provides greater flexibility for use in predictions, which leads to better video compression. The larger block size is now available. The 16 by 16 and the 8 by 16 provide efficient coding for uniformly predictable regions in the pictures. The numbers in the green and the gray blocks represent the order in which each block of the macroblock pair is processed.



ECF No. 160-2, Ex. K, Fig. 17

§6,1409-1445 Now, if a picture consists of some regions that are moving and some regions that are not, it is typically more efficient to code the nonmoving regions in frame mode and the moving regions in field mode. Using macroblock pairs gives you the flexibility to frame code nonmoving regions and field code moving regions while still using all seven block sizes for prediction. Here, I show a picture of one vertical pair of macroblocks that is frame coded and one vertical pair that is field coded.



200

FIG. 7

§6,1446-1479 A frame/field flag is included in the bitstream before each pair. This flag indicates whether the subsequent pair of macroblocks in the bitstream is coded in frame mode or in field mode.

Frame Or Field, Is Used In Coding

> In AFF coding at the macroblock level, a frame/field flag bit is preferably included in a picture's bitstream to indicate which mode, frame mode or field mode, is used in the encoding of each macroblock. The bitstream includes information pertinent to each macroblock within a stream, as shown in FIG. 11. . . . If the AFF is performed on pairs of macroblocks, the frame/field flag (112) is preferably included before each pair of macroblock in the bitstream.
>
> '374 Patent, 8:46-58

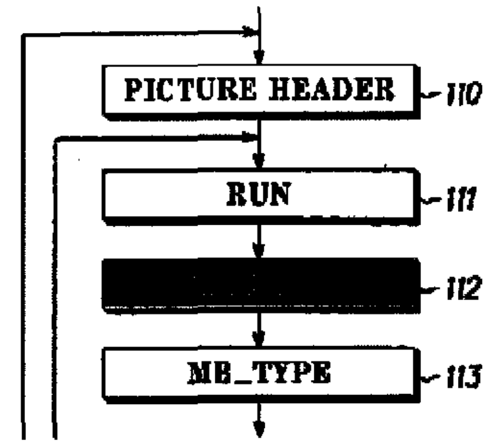
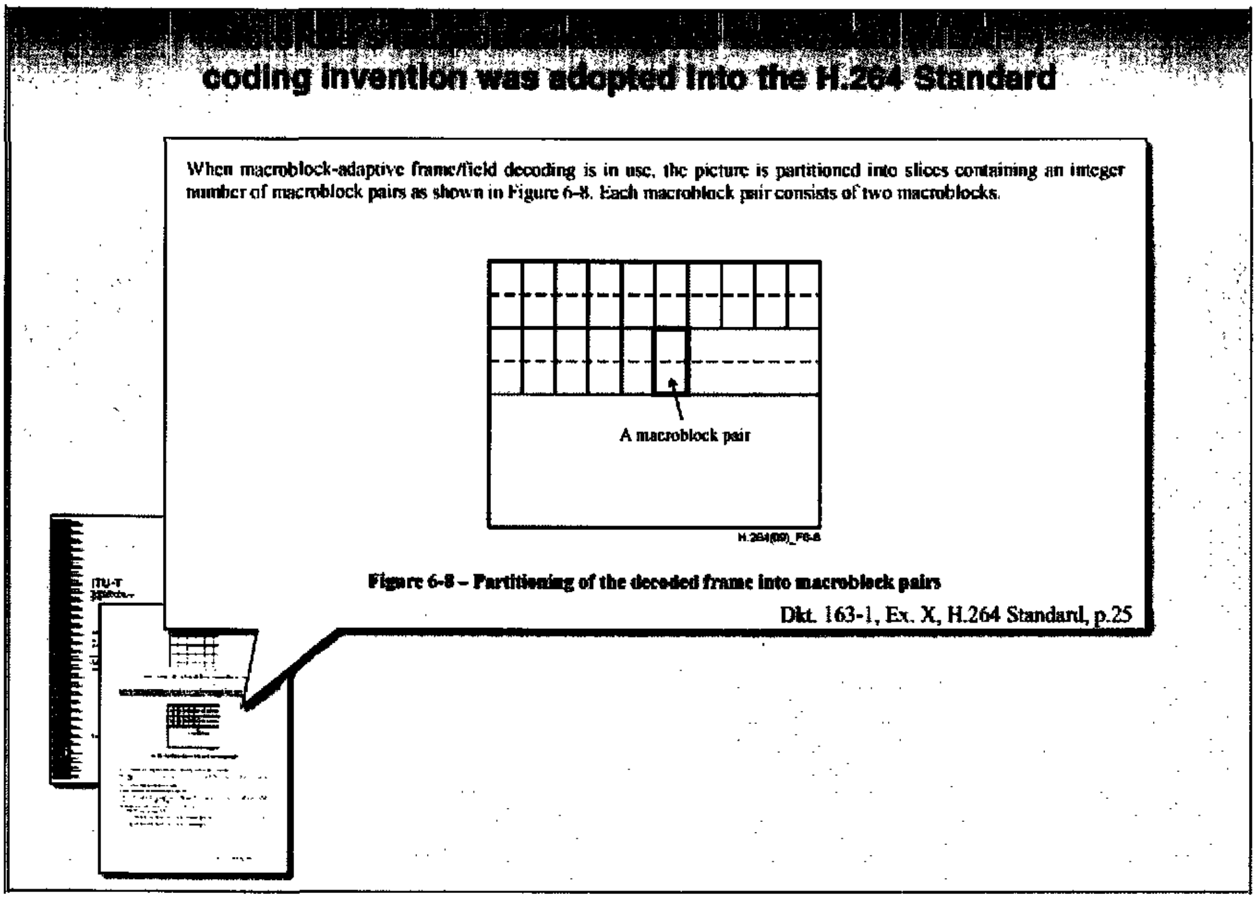


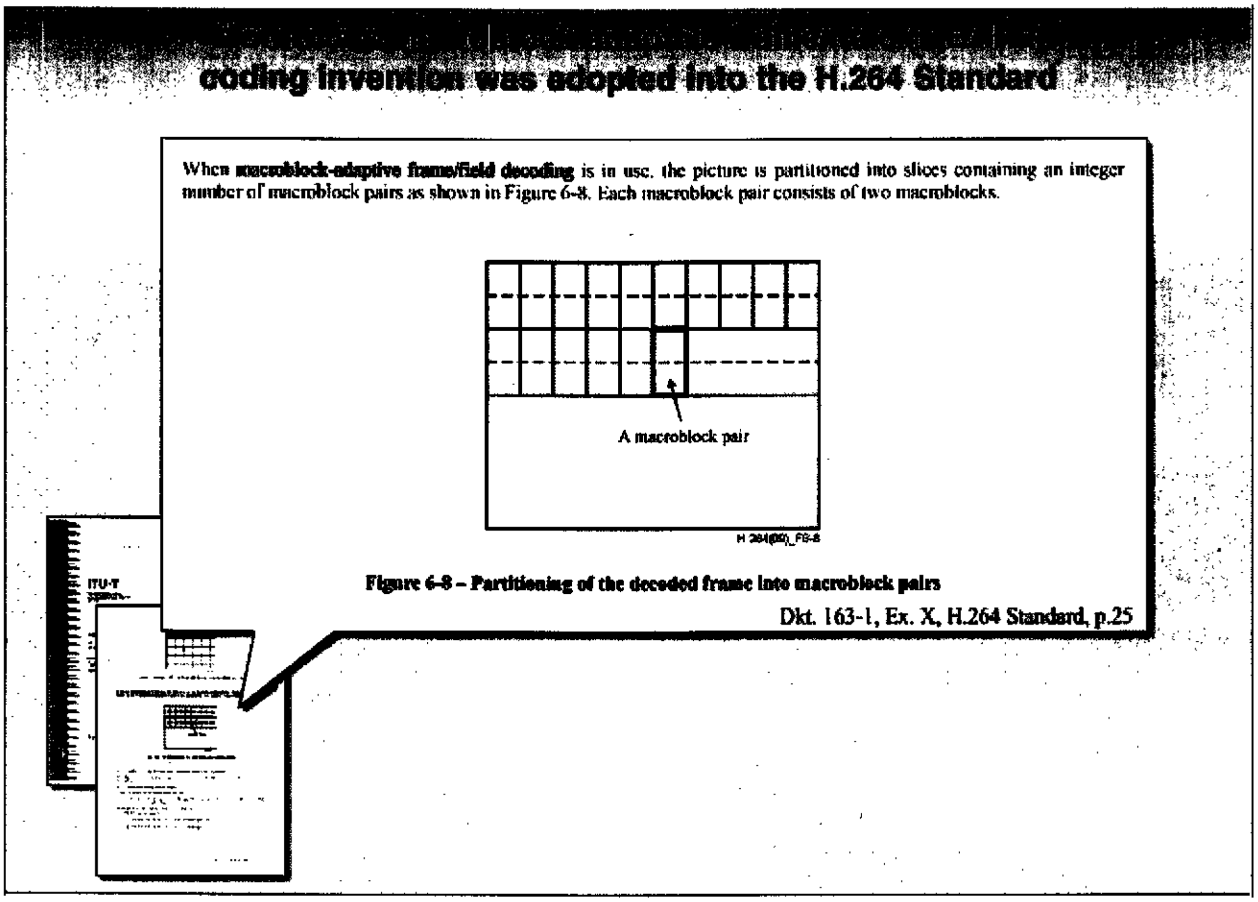
FIG. 11

29682505_10

§7 Adaptive frame/field coding on macroblock pairs was adopted by the video coding experts into the H.264 standard. For example, Figure 6-8 of the H.264 standard shows partitioning of a frame into macroblock pairs.



§7,1 The H.264 standard refers to the use of macroblock pairs as macroblock adaptive frame/field coding or MBAFF.



§8 Each of the Motorola patents is directed to perform that coding on pairs or groups of macroblocks. In addition, each patent further entails either inter-coding, the '374 patent; intra-coding, the '375 patent; or a particular scanning path, the '376 patent.

The '374, '375 and '376 Patents

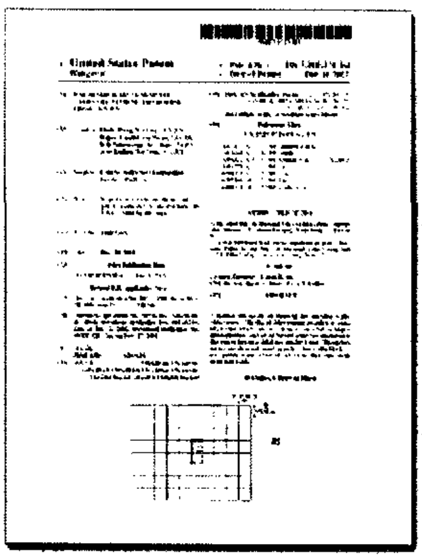
The Asserted Patents are directed to Motorola's MBAFF Invention together with:

- **Inter coding ('374)**
- **Intra coding ('375)**
- **Scanning Paths ('376)**

§9 First, I would like to discuss inter-coding. In inter-coding, macroblocks are coded by first looking for part of a different picture that is very similar to what we are trying to encode, and which we already know. That is called the reference picture. Then we compute the difference between the two parts, which you call the residual, and restore or transmit only that. Inter-coding is also referred to as temporal prediction with motion compensation.

Invention Together With Inter Coding

Inter coding is coding a picture by referring to reference pictures (*i.e.*, pictures earlier or later in time) in order to predict the values of a block of pixels.



"According to an embodiment of the present invention, each frame and field based macroblock in macroblock level AFF can be intra coded or ***inter coded***. ... On the other hand, in ***inter coding***, [redacted]

'374 Patent, 9:9-15

29682505_10

§10,1 As I explained earlier, a digital video is a sequence of pictures, as shown in Figure 1 of the patents.

[illegible]
By Referring To A Similar Macroblock In A Reference Picture

> A predicted picture (102a, b) is encoded using an I, P, or B picture that has already been encoded as a reference picture.
>
> '374 Patent, 5:16-18

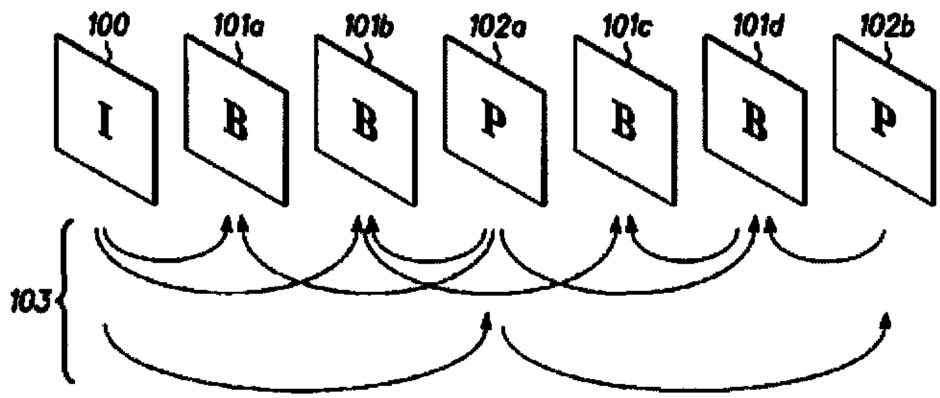
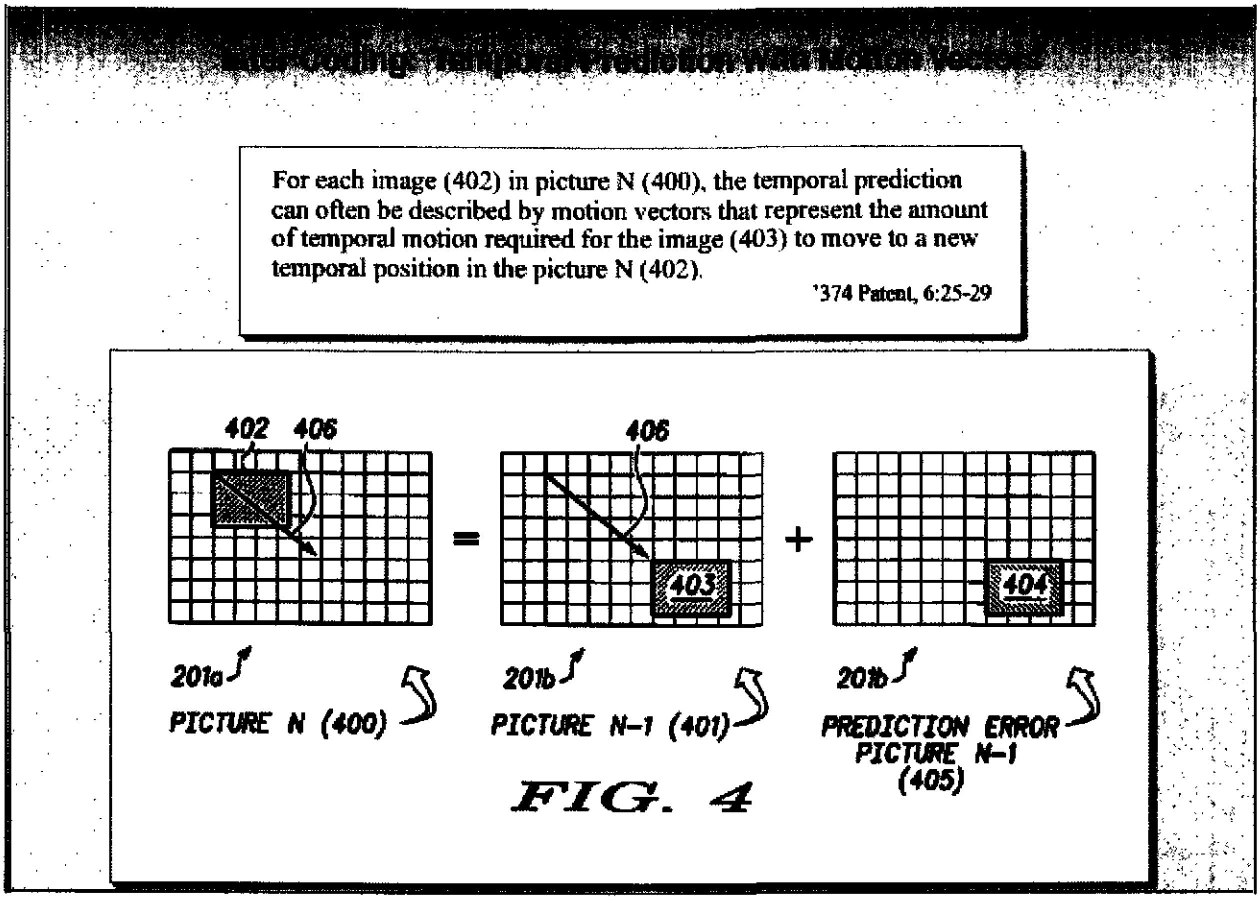


FIG. 1

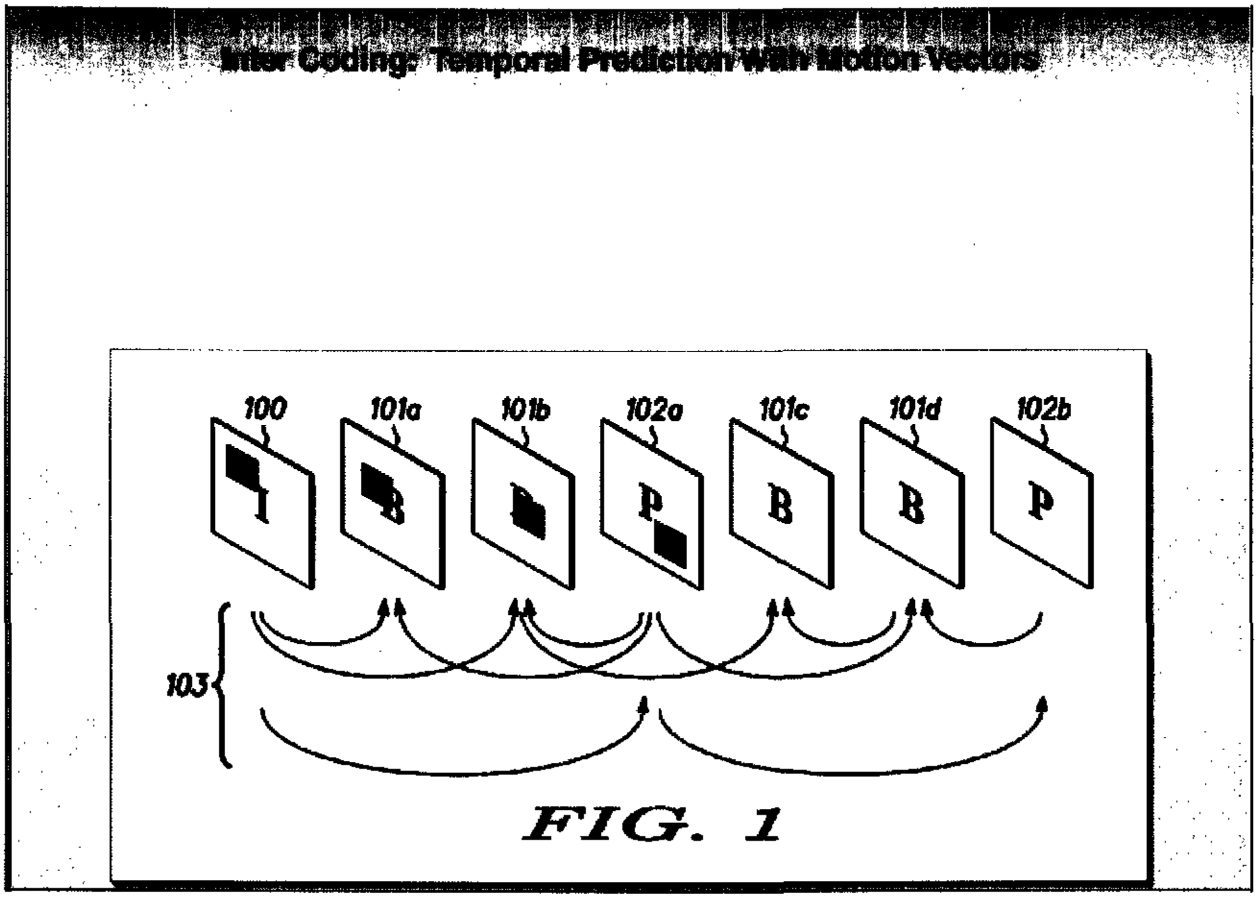
29682505_10

§10,2 The '374 patent gives an example of inter-coding in Figure 4. In this example, we generate Image 402, in a picture based on the corresponding Image 403 in the reference picture.

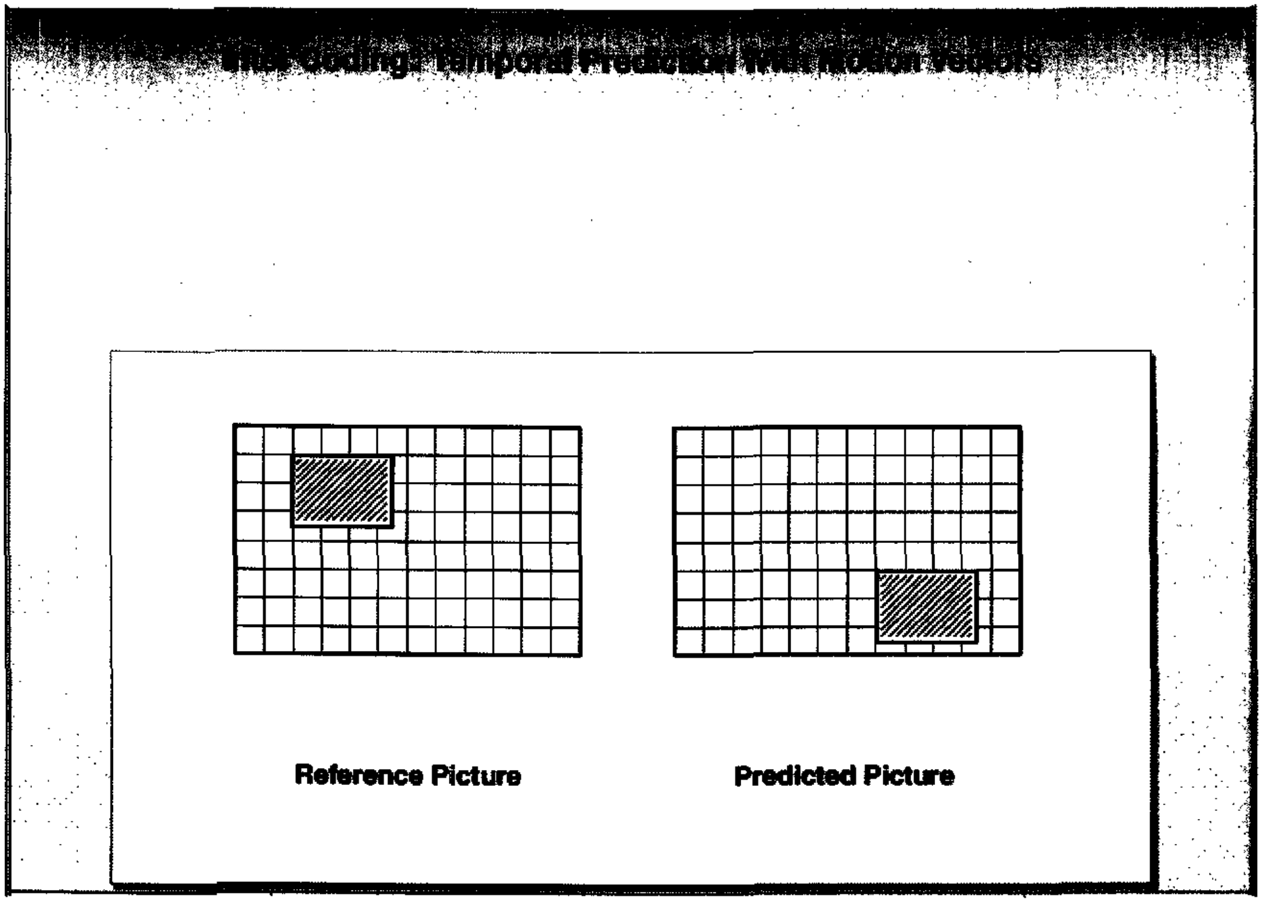


29682505_10

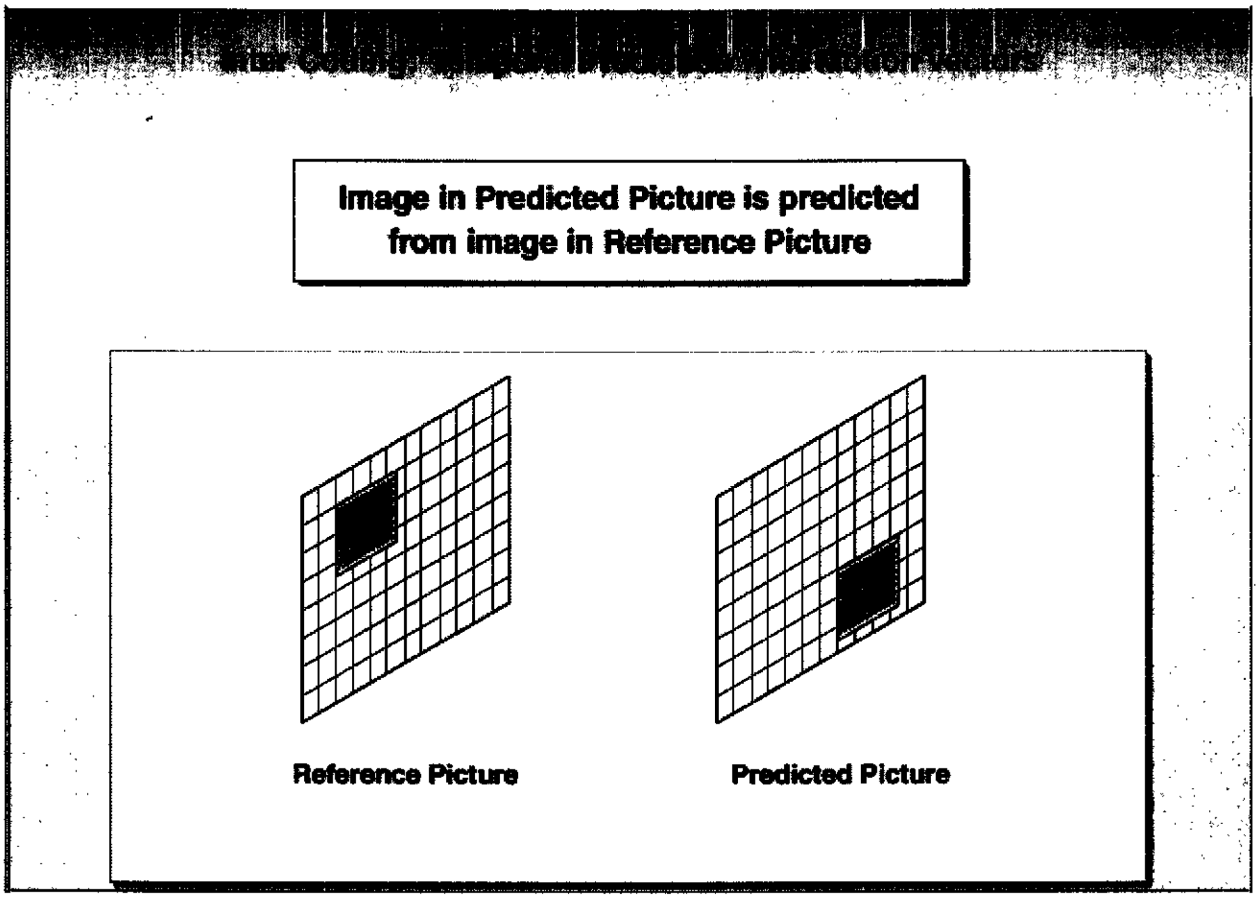
§10,2-70 As objects move across a camera's field of view, they show up in different parts of the picture as time passes. Here, I illustrate a rectangular in red moving from upper left to lower right as time passes.



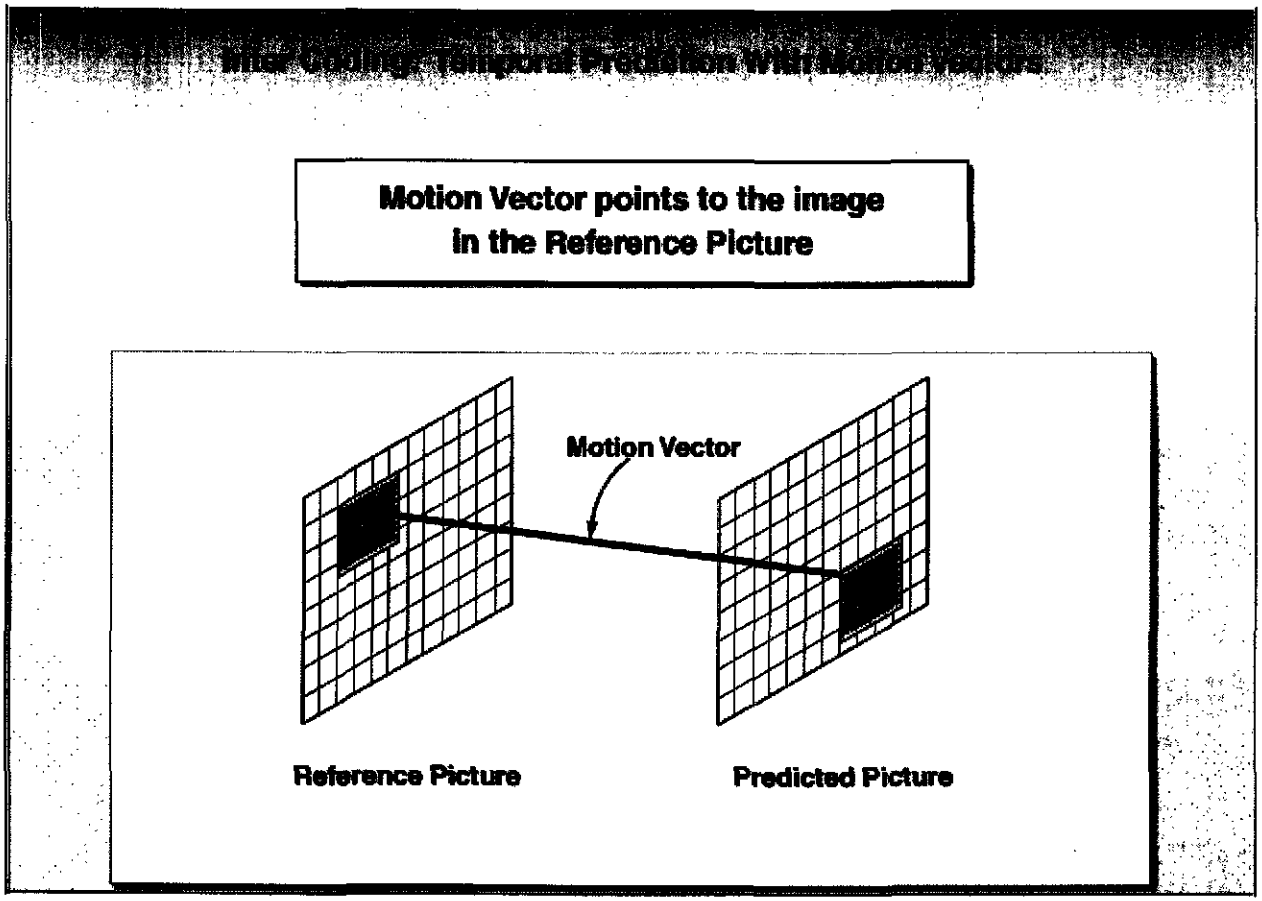
§10,71-169 We can predict the P picture from the I picture. We refer to the I picture as a reference picture and the P picture as the predictive picture.



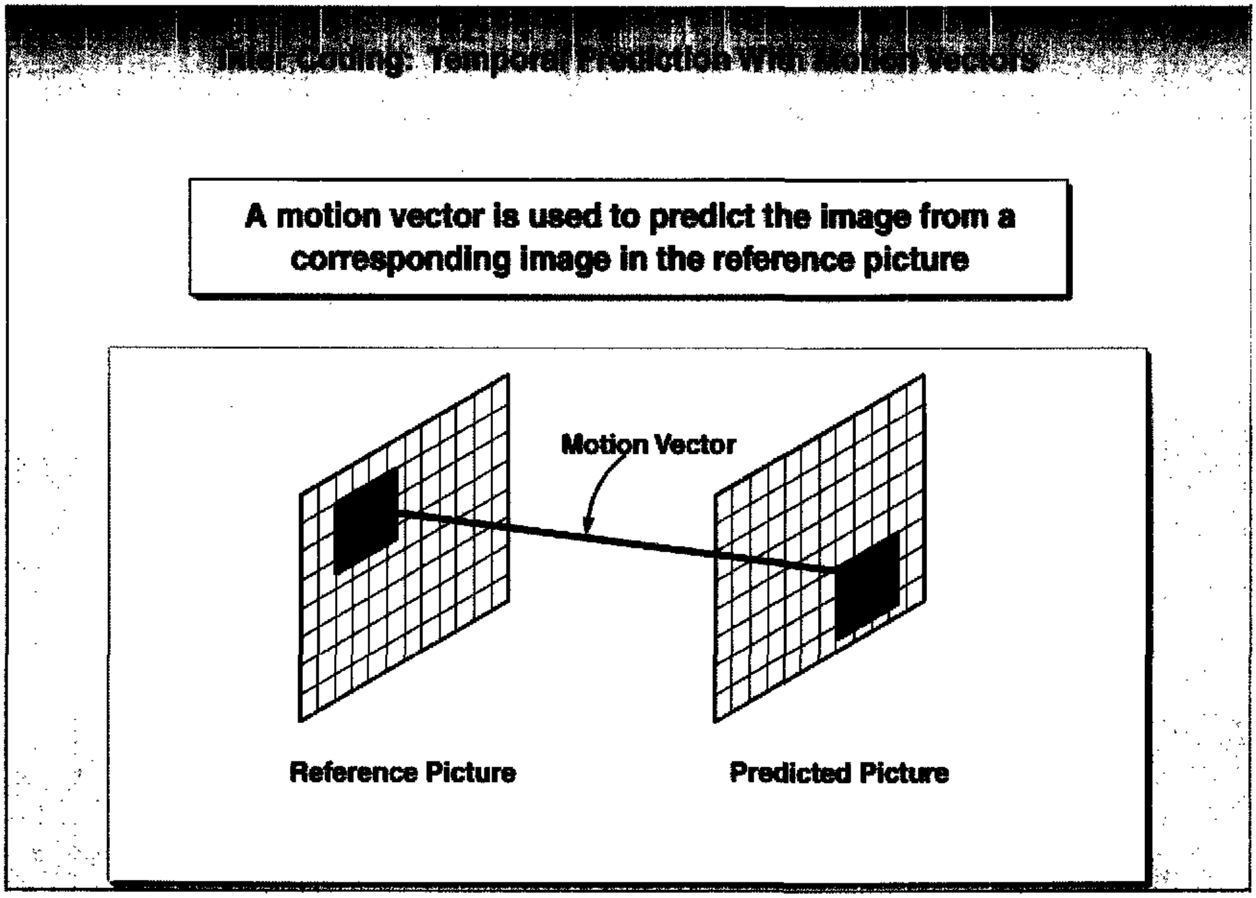
§10,170-210 I will put this into three dimensions so it is easier to see what is happening here.



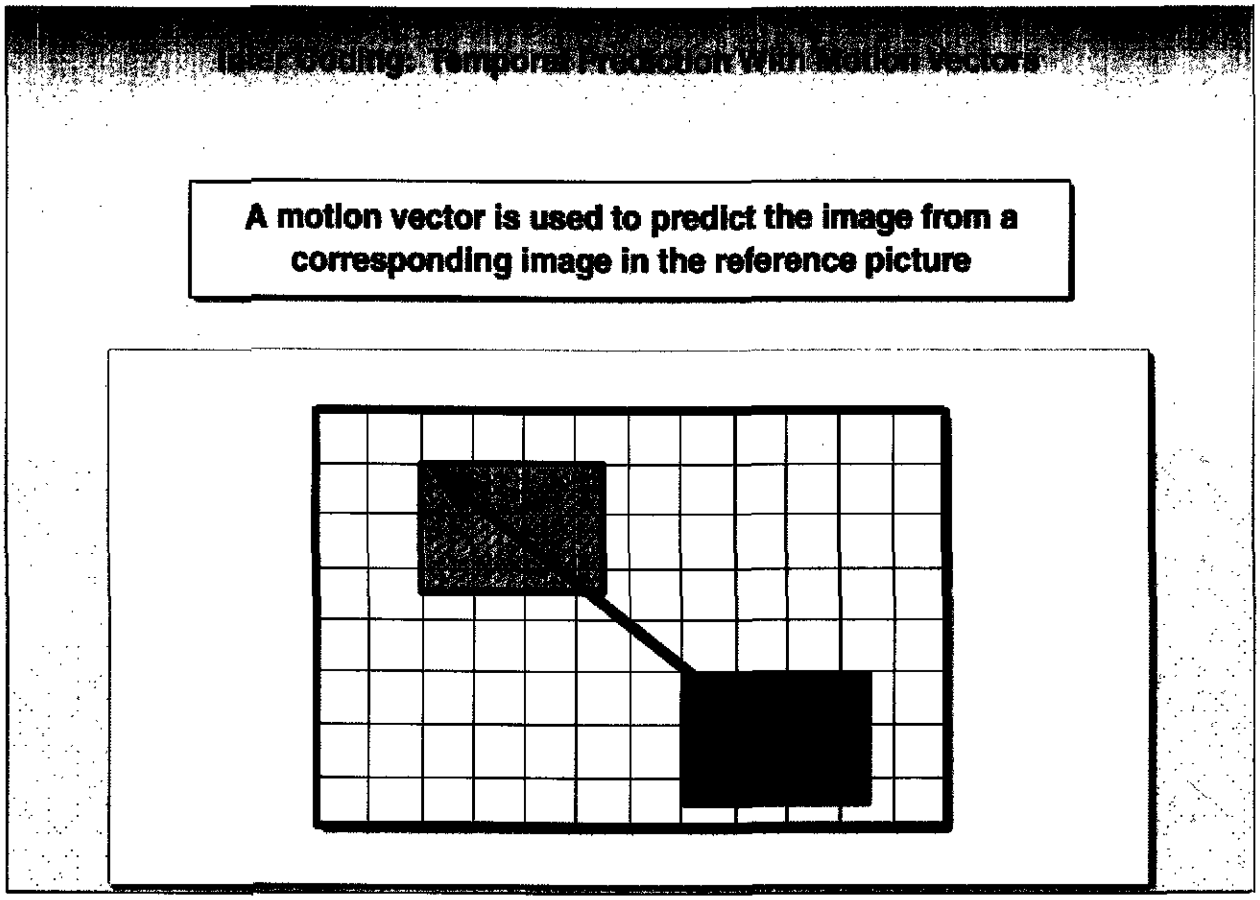
§10,211-242 In order to encode an image in the predicted picture, the encoder reads in the motion vector that tells us where to find a related image of predictor in the reference picture.



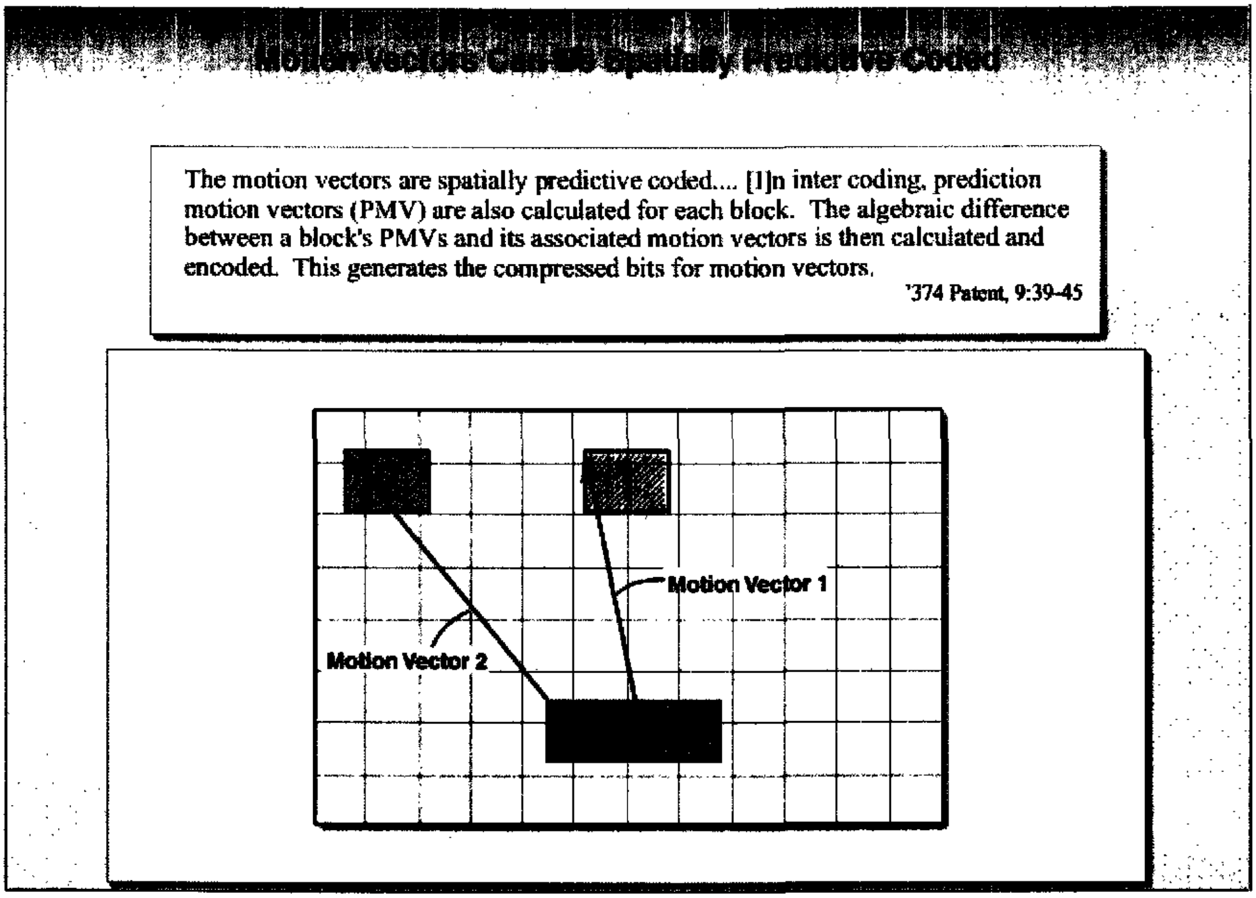
§10,243-325 The image in the reference picture is used to create the image in the predicted picture. The motion vector tells us where to find the predictor.



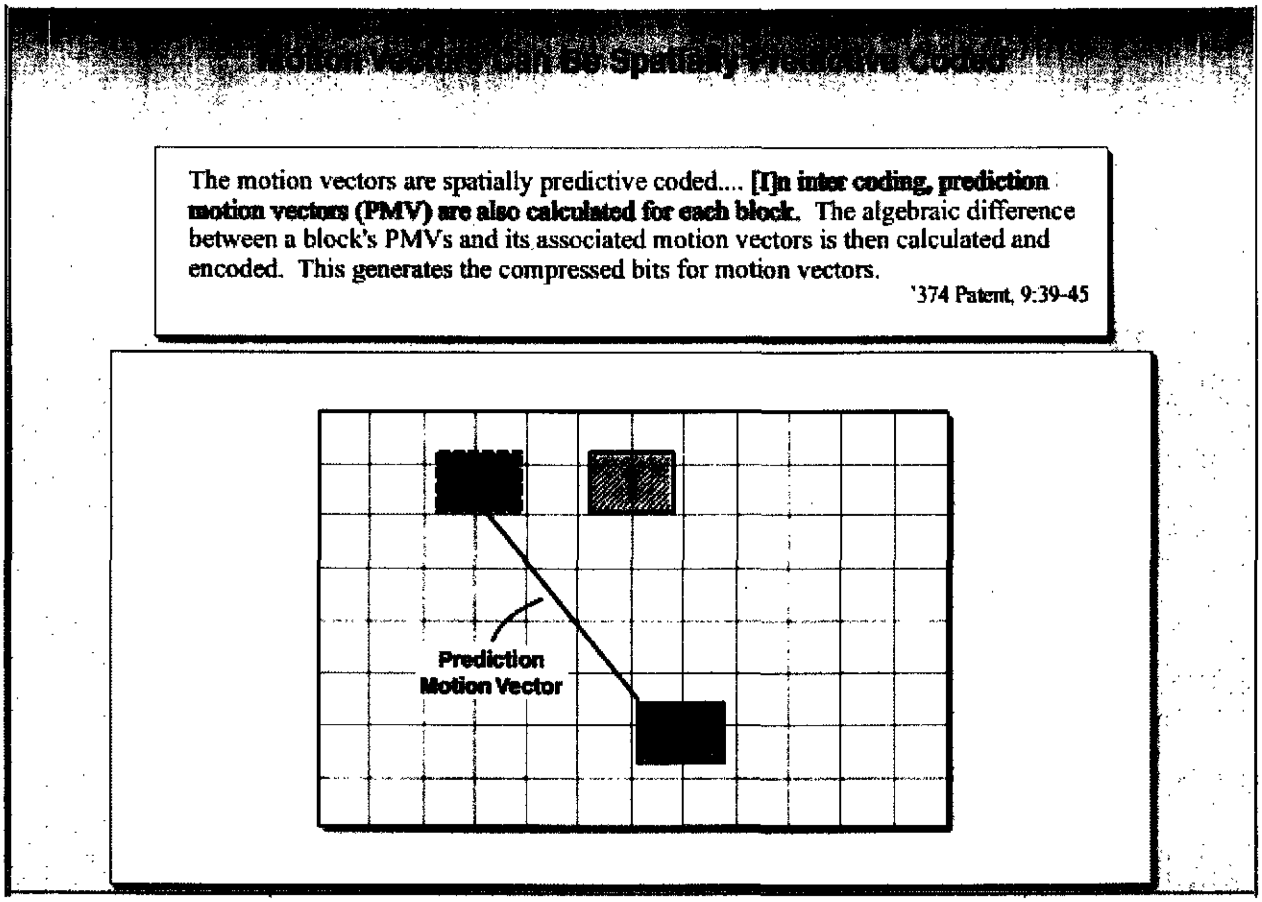
§10,326-584 By overlaying the reference picture atop the predicted picture, we can see how this process looks in two dimensions. We are trying to encode this picture. We find a predictor in a reference picture, and we have a motion vector that tells us how to get there.



§11,1 A motion vector can also be compressed. A motion vector is compressed by predicting it based on another block's motion vector. Here, I will illustrate two blocks, Block 1 and Block 2. The bold blocks represent where the blocks are in the current image, and the faded blocks represent where they are in the reference picture. Block 1, moved by the amount shown by Motion Vector 1, and Block 2, moved by the amount shown by Motion Vector 2, except with the arrows in the opposite sense. So Motion Vectors 1 and 2 are different, because Blocks 1 and 2 move by different amounts.

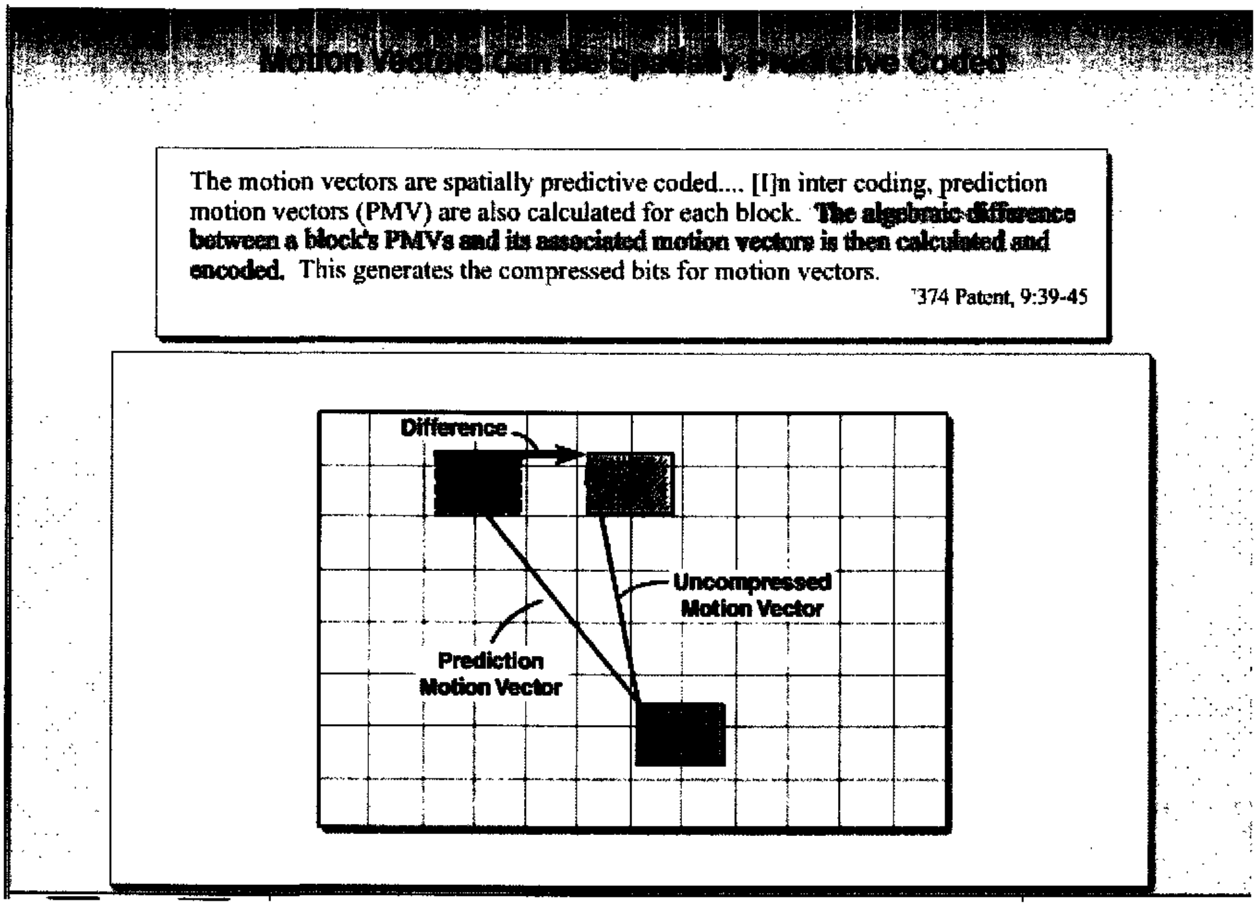


§11,2-222 We can predict Motion Vector 1 based on Motion Vector 2. In that case, Motion Vector 2 becomes the prediction motion vector for Block 1. However, if we use only the prediction motion vector to predict the position of Block 1 in the current picture, we see that it points too far to the left. That is the hashed version of Block 1 that I have shown.



29682505_10

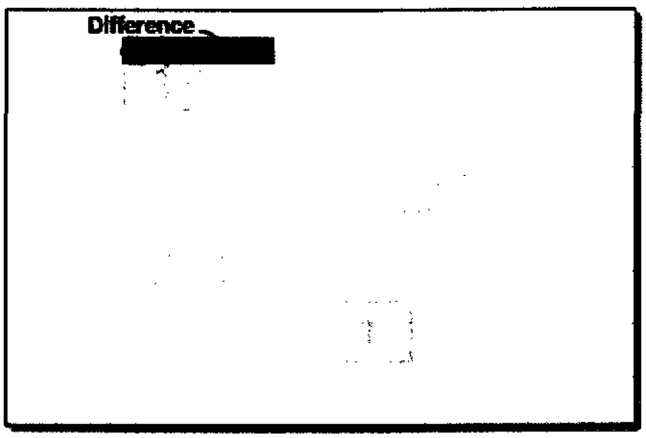
§11,223-283 Because it points too far to the left, we calculate a difference value, which I have shown as a green arrow. This green arrow represents the difference between Block 1's prediction of the motion vector, which was based on Block 2, that's here, and the uncompressed motion vector for Block 1.



§11,284-305 The difference value is a compressed version of Block 1's motion vector. Only the difference value needs to be coded. And that's what is transmitted in the bitstream. When decoding the predicted picture, the difference value is used to generate the uncompressed motion vector for the block. And that is all I have to say about inter-coding.

Motion Vectors Can Be Spatially Predictive Coded

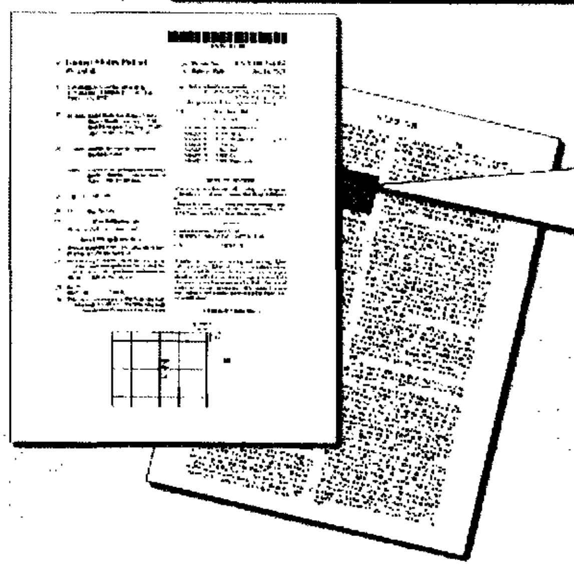
> The motion vectors are spatially predictive coded.... [I]n inter coding, prediction motion vectors (PMV) are also calculated for each block. The algebraic difference between a block's PMVs and its associated motion vectors is then calculated and encoded. **This generates the compressed bits for motion vectors.**
>
> '374 Patent, 9:39-45



§12 We need to talk about intra-coding. In intra-coding, macroblocks are predicted based on the pixel values of their neighboring blocks within the same picture. Intra-coding is also referred to as spatial prediction. That means we don't have to use a different reference picture.

The '374 Patent Claims Are Directed To The MBAFF Invention Together With Intra Coding

Intra coding is coding a picture without referring to reference pictures (*i.e.*, pictures earlier or later in time) by predicting the values of a block of pixels based on pixels from neighboring blocks within the same picture.



"According to an embodiment of the present invention, each frame and field based macroblock in macroblock level AFF can be ***intra coded*** or ***inter coded***. In intra coding, the macroblock is encoded [redacted]"

'374 Patent, 9:9-13

29682505_10

§13 Because the code pictures, starting from the upper left corner and proceeding left to right, from top to bottom, the only pixels from the neighboring blocks that we know are those that are above and to the left of what we are immediately trying to encode. *And this is illustrated in Figure 14 of the patents.*

Pixels In Neighboring Blocks

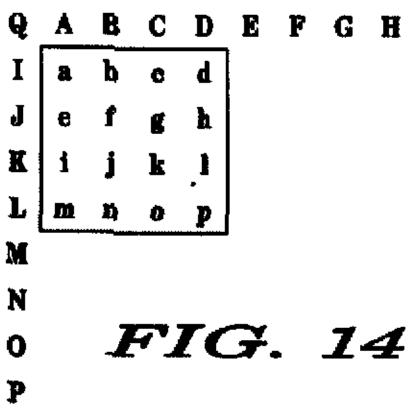


FIG. 14

"For intra$_{4x4}$ mode, the ***predictions of the pixels*** in a 4 by 4 pixel block, as shown in FIG. 14, are derived form its ***left and above pixels***. In FIG. 14, the 16 pixels in the 4 by 4 pixel block are labeled a through p. Also shown in FIG. 14 are the neighboring pixels A through P. The neighboring pixels are in capital letters."

'374 Patent, 14:46-51

29682505_10

§14 Intra-coding can be performed in any of nine directions as shown in Figure 15. I will give the vertical direction zero as an example.

[illegible]
Pixels In Neighboring Blocks

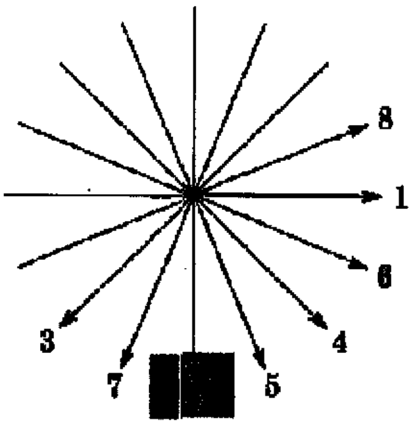


FIG. 15

"As shown in FIG. 15, there are ***nine different prediction directions*** for intra_{4x4} coding. They are [redacted], horizontal (1), DC prediction (mode 2), diagonal down/left (3), diagonal down/right (4), vertical-left (5), horizontal-down (6), vertical-right (7), and horizontal-up (8). DC prediction averages all the neighboring pixels together to predict a particular pixel value."

'374 Patent, 14:51-57

§15,1,2-212 For intra-prediction in the vertical prediction, the value of the above neighbor pixel, capital A, is copied downwards in the vertical direction into A, E, I and M. Similarly, B is copied into B, F, J and M. Likewise, C is copied into C, G, K and O. And again with D. So the result is a 4 by 4 block of pixels that approximates the block being coded.

Intra-prediction works similarly in the horizontal and other directions. And we can think of this as taking a brush loaded with the tone values of from A, B, C, D and across, and sweeping the brush downward to create our predictor. For the other directions, we just sweep the brush in other directions. That concludes my discussion of intra-coding.

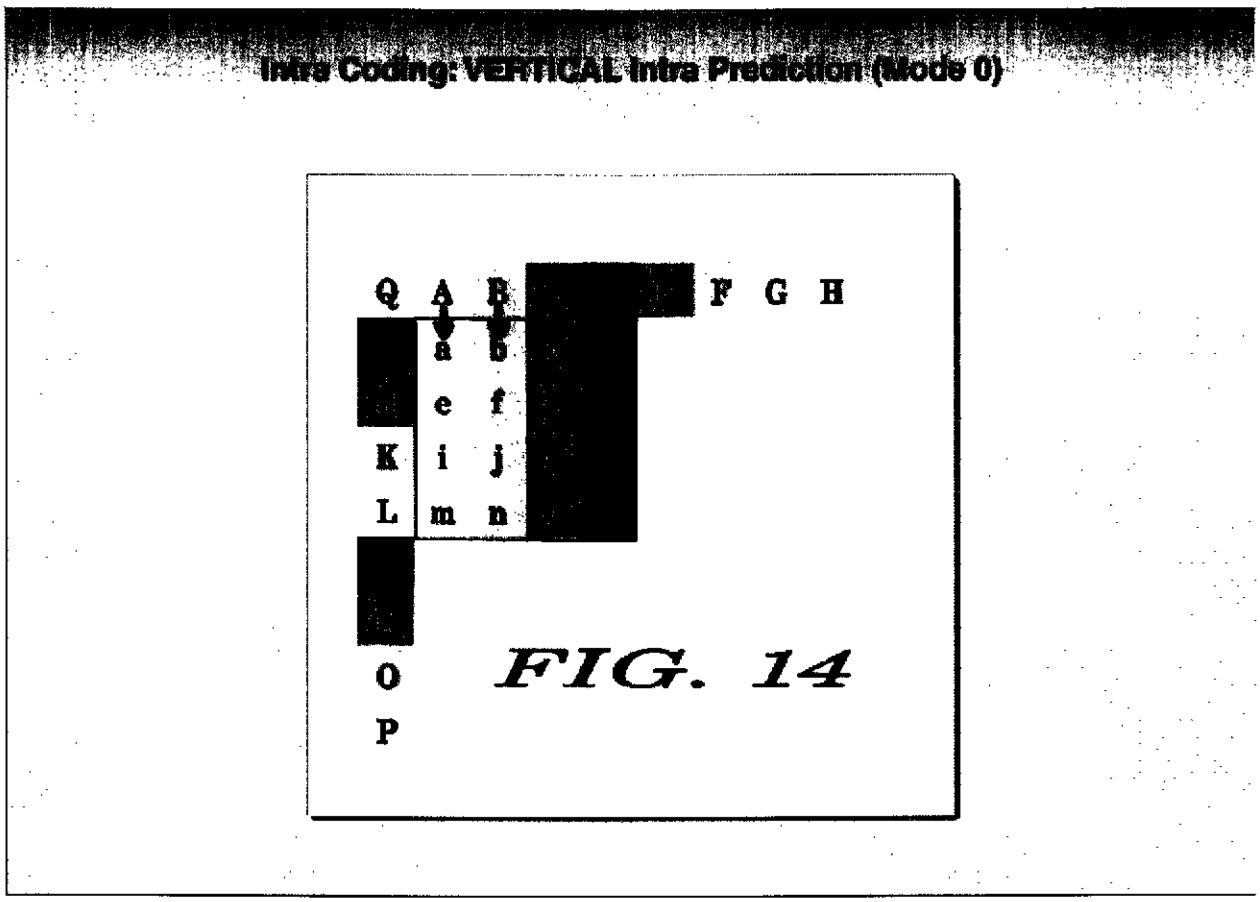
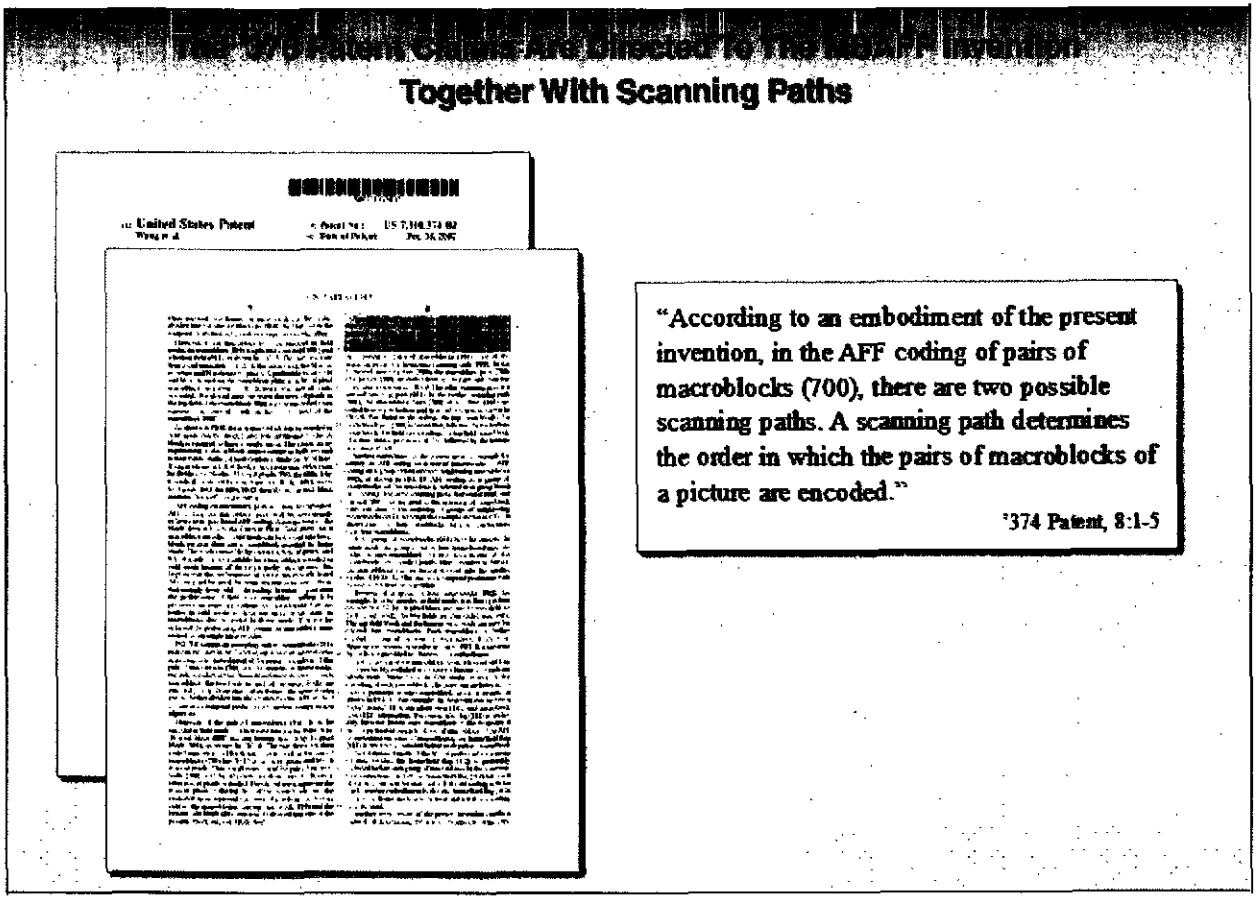


FIG. 14

§16 Finally, I would like to discuss scanning paths. The scanning path determines the order in which macroblocks are processed. When AFF coding pairs a group of macroblocks, there are two possible ways in which to scan macroblock pairs or groups within a picture, vertical or horizontal.



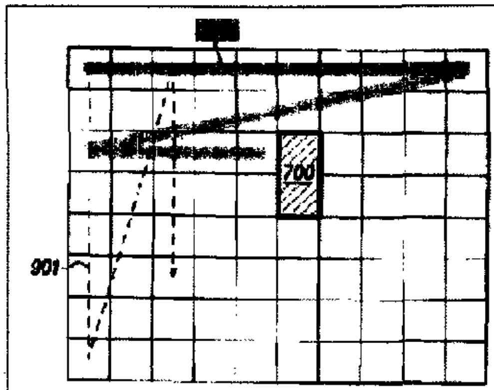
29682505_10

§17 Figure 9 depicts horizontal and vertical scanning paths for a picture that has been partitioned into pairs of macroblocks. Both scanning paths start at the upper left corner of the picture. The horizontal scan path 900 scans each row of macroblock pairs from left to right. As shown in Figure 9, the second row is skipped because vertically adjacent pairs are scanned together.

Two Possible Scanning Paths: Horizontal And Vertical

"FIG. 9 shows the two possible scanning paths in AFF coding of pairs of macroblocks (700). One of the scanning paths is a horizontal scanning path (900). In the **horizontal scanning path (900)**, the macroblock pairs (700) of a picture (200) are coded from left to right and from top to bottom, as shown in FIG. 9. The other scanning path is a vertical scanning path (901). In the vertical scanning path (901), the macroblock pairs (700) of a picture (200) are coded from top to bottom and from left to right, as shown in FIG. 9."

'374 Patent, 8:5-14



200 *FIG. 9*

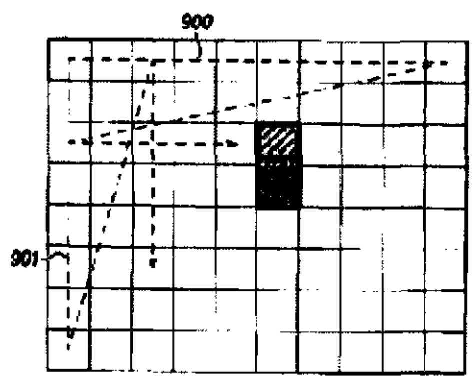
29682505_10

§18 Within a macroblock pair, the top macroblock is encoded first, followed by the bottom macroblock. That concludes my discussion of scanning paths and of Motorola's technology tutorial. Thank you, your Honor, for your kind attention and patience. I would be happy to answer any questions you may have.

Horizontal And Vertical Scanning Paths For A Macroblock Pair

"For ***frame mode coding***, the **top macroblock of a macroblock pair (700)** is coded first, followed by the [redacted]. For ***field mode coding***, the **top field macroblock** of a macroblock pair is coded first followed by the [redacted]"

'374 Patent, 8:14-18



200 FIG. 9

29682505_10

§19 THE END.

This concludes the video coding technology tutorial submitted by Motorola.